September, 2021

WorldFish Carp Genetic Improvement Program Data Management System (Version 3): Data input

Matthew Hamilton

Citation

This publication should be cited as: Hamilton MG. 2021. WorldFish Carp Genetic Improvement Program Data Management System (Version 3): Data input. Penang, Malaysia: WorldFish. Program Report.

About WorldFish

WorldFish is a nonprofit research and innovation institution that creates, advances and translates scientific research on aquatic food systems into scalable solutions with transformational impact on human well-being and the environment. Our research data, evidence and insights shape better practices, policies and investment decisions for sustainable development in low- and middle-income countries.

We have a global presence across 20 countries in Asia, Africa and the Pacific with 460 staff of 30 nationalities deployed where the greatest sustainable development challenges can be addressed through holistic aquatic food systems solutions.

Our research and innovation work spans climate change, food security and nutrition, sustainable fisheries and aquaculture, the blue economy and ocean governance, One Health, genetics and AgriTech, and it integrates evidence and perspectives on gender, youth and social inclusion. Our approach empowers people for change over the long term: research excellence and engagement with national and international partners are at the heart of our efforts to set new agendas, build capacities and support better decision-making on the critical issues of our times.

WorldFish is part of One CGIAR, the world’s largest agricultural innovation network.

Acknowledgments

This work was undertaken as part of the CGIAR Research Program on Fish Agri-Food Systems (FISH) led by WorldFish. The program is supported by contributors to the CGIAR Trust Fund.

Contact

WorldFish Communications and Marketing Department, Jalan Batu Maung, Batu Maung, 11960 Bayan Lepas, Penang, Malaysia. Email: worldfishcenter@cgiar.org

Creative Commons License

Photo credits

Front cover, Matthew Hamilton.

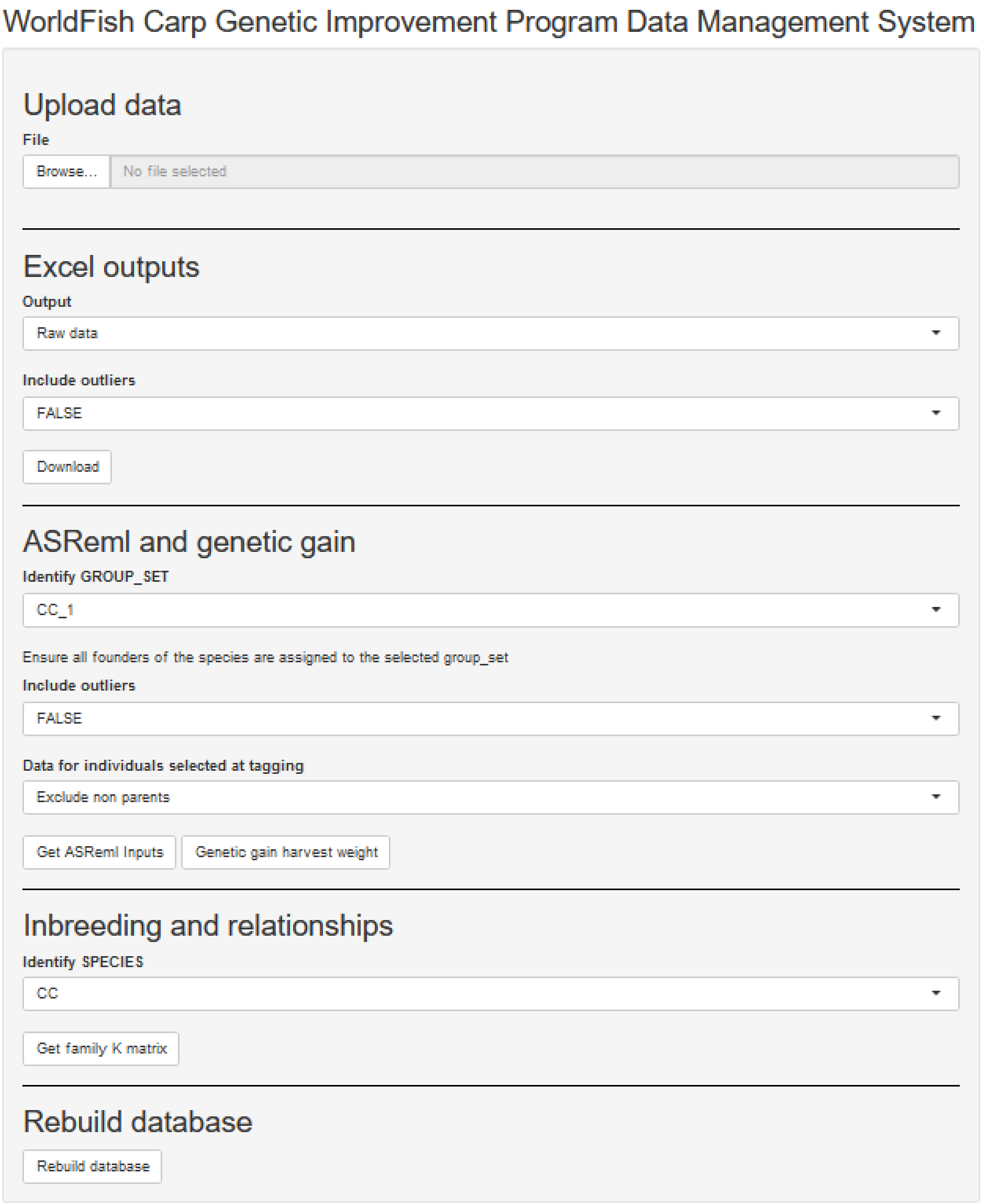
Table of contents

1. Introduction

The WorldFish Carp Genetic Improvement Program (CGIP) is comprised of three species: catla (*Catla catla*), rohu (*Labeo rohita*) and silver carp (*Hypophthalmichthys molitrix*) (Hamilton et al., 2019a; Hamilton et al., 2019b; Hamilton et al., 2021) and generates substantial volumes of data each year. Core CGIP genetic and measurement data are retained in a SQLite relational database (‘the CGIP Database’; Annex 1) (Müller et al., 2021) – a separate system is used to store CGIP pond environment and pond management data (Hamilton et al., 2020). The CGIP Database aims to store data in an accurate and consistent manner, to allow complete or partial automation of error-checking, data validation, data analysis and data reporting procedures via the CGIP Data Management System or tailored R scripts (R Core Team, 2020).

This report details the process of data input to the CGIP Database through the CGIP Data Management System Graphical User Interface (Chang et al., 2021) (Figure 1). Outputs of the CGIP Data Management System are not described. Access to the CGIP Data Management System and GCIP Database is available to authorised WorldFish staff only.

WorldFish CGIP data is loaded to the GCIP Database as soon as possible after collection. This allows any errors to be captured quickly and minimises the possibility of data loss. Load (i.e. data input) procedures within the CGIP Data Management System are deliberately rigid in terms of the order in which data can be loaded (see Section 3) and the format of load files (see Section 4). In brief, after a relevant CGIP activity has been completed (Figure 2; Tables 1 and 2), the first step to input data is to generate a load file (Figure 3; Table 3; Section 4). Load files have a defined set of case-sensitive fields and must be saved as comma-delimited (.csv) files prior to loading. To allow the CGIP Data Management System to identify the type of data in the file, load files must be saved with a name that includes one, and only one, keyword (Table 3). Once the load file is saved, it is necessary to click on the ‘Browse…’ button in the ‘Upload data’ section of the CGIP Data Management System Graphical User Interface (Figure 1) and select the relevant comma-delimited file. If an error is present in the load file, a message will be returned describing the error and the affected records. The error must then be rectified in the load file, before repeating the load process. If there are multiple errors in a load file, this process may need to be repeated multiple times. Only once the load file is error free is the data loaded to GCIP Database by the CGIP Data Management System.



WorldFish Carp Genetic Improvement Program Data Management System

Upload data

File

No file selected

Excel outputs

Output

Raw data

Include outliers

FALSE

ASReml and genetic gain

Identify GROUP_SET

CC_1

Ensure all founders of the species are assigned to the selected group_set

Include outliers

FALSE

Data for individuals selected at tagging

Exclude non parents

Inbreeding and relationships

Identify SPECIES

CC

Rebuild database

Figure 1. Graphical user interface of the WorldFish Carp Genetic Improvement Program Data Management System.

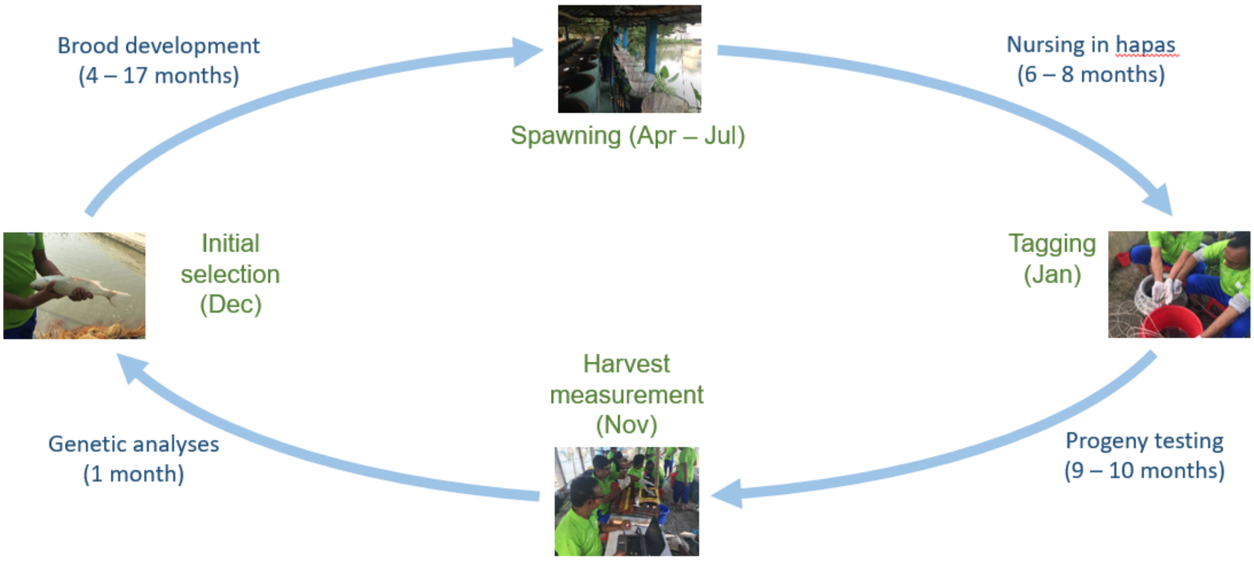


Figure 2. WorldFish Carp Genetic Improvement Program selective breeding cycle.

Activity	Description
Spawning	New families generated from tagged parents.
Nursing	Describe family nursing environment.
Tagging / family assignment	Tagging of fish with individual identifiers and/or assignment of individuals to families.
Progeny testing	Tagged fish of known pedigree progeny tested.
Measurement	Measurement of fish for defined traits.
Genetic analysis	Updated estimated breeding values (EBVs) computed.
Death	Death/culling of tagged fish.

Table 1. Activities in the selective breeding cycle that trigger generation of a load file and data entry into the WorldFish Carp Genetic Improvement Program Database.

Activity	Description
Data checking	Data loaded to the GCIP Database checked for errors.
Database described	Relationships among GCIP Database tables described and data sources defined
Dissemination	Fish released or disseminated to external parties/partners.
Founders introduced	Describe founders. Founders are fish sourced from outside the CGIP with no known familial relationship with exiting fish or other founders.
Tissue sampling	Tissue samples obtained from tagged individuals.

Table 2. Miscellaneous activities that trigger generation of a load file and data entry into the WorldFish Carp Genetic Improvement Program Database.

Load file keyword	Description	Triggering activity	Dependencies	Database tables populated
aggregate_desc	Describes aggregates.	‘Dissemination’	‘spawning’	‘AGGREGATE_DESC’
aggregate_fam	Assigns families to aggregates.	‘Dissemination’	‘aggregate_desc’ and ‘spawning’	‘AGGREGATE_FAM’
death	Specifies date of death for individuals.	‘Death’	‘tagging’	‘DEATH’
ebv_data	Loads estimated breeding values (EBVs).	‘Genetic analysis’	‘ebv_run’, ‘ebv_treat’ and ‘tagging’	‘EBV’
ebv_run	Identifies unique estimated breeding value (EBV) run (i.e. computation).	‘Genetic analysis’	‘species’	‘EBV_RUN’
ebv_treat	Identifies unique combinations of trait and treatment for which estimated breeding values (EBVs) may be computed.	‘Genetic analysis’	‘species’, ‘trait’, and ‘treatment’	‘EBV_TRAIT_TREAT’
family_assign	Assigns a family to an individual if the family was unknown at the time of tagging.	‘Tagging / family assignment’	‘tagging’	‘INDIV_FAM’
feed	Provides details of feeds	‘Genetic analysis’ and ‘Progeny testing’	None	‘FEED’
fert	Provides details of fertilisers	‘Genetic analysis’ and ‘Progeny testing’	None	‘FERT’
founder_desc	Assigns founders to species and origin	‘Founders introduced’	‘origin’ and ‘spawn_round’	‘INDIV_PIT’, ‘INDIV_SPECIES’ and ‘FOUNDER’
group_define	Defines genetic groups within a previously defined ‘group set’.	‘Founders introduced’	‘group_set’	‘GROUP_DEFINE’
group_founder	Assigns founders to genetic groups.	‘Founders introduced’	‘group_define’ and ‘founder_desc’	‘GROUP_FOUNDER’
group_set	Defines a ‘set’ of genetic groups. Each founder can be assigned to a genetic group in multiple genetic group ‘sets’.	‘Founders introduced’	‘species’	‘GROUP_SET_DESC’

Load file keyword	Description	Triggering activity	Dependencies	Database tables populated
hapa	Defines a nursing hapa (or chamber/pond) identifier within a spawn round	‘Nursing’	‘spawn_round’	‘HAPA’
indiv_stock	Assigns individuals to a grow-out pond.	‘Progeny testing’	‘hapa’, ‘pond_desc’ and ‘tagging’	‘INDIV_STOCK’
measure_data	Loads measurement data.	‘Measurement’	‘measure_event’, ‘tagging’, and ‘trait’	‘MEASURE_DETAILS’ and ‘MEASURE’
measure_event	Defines measurement ‘events’. All measurement data must ultimately be assigned to a measurement event.	‘Measurement’	None	‘MEASURE_EVENT’
nurse_count	Enters the count of individuals surviving in each hapa at the time hapa transfer and tagging	‘Nursing’	‘hapa’	‘NURSE_COUNT’
nurse_details	Enters details of hapa positions, mesh size and movements during nursing.	‘Nursing’	‘hapa’ and ‘pond_desc’	‘NURSE_DETAILS’
origin	Defines origins of founders (e.g. river and coordinates, hatchery). All founders must be assigned to an origin.	‘Founders introduced’	‘species’	‘ORIGIN’
group_define	Defines genetic groups within a previously defined ‘group set’.	‘Founders introduced’	‘group_set’	‘GROUP_DEFINE’
outlier	Identifies outliers in measurement data.	‘Data checking’	‘measure_data’	‘OUTLIER’
partner	Details of partners (e.g. hatcheries and nurseries).	‘Dissemination’	None	‘PARTNER’
pond_desc	Identifies ponds within sites.	‘Nursing’ and ‘Progeny testing’	‘site’	‘POND’.
pond_treat	Describes progeny test (i.e. grow-out pond) treatments for each spawn round.	‘Progeny testing’	‘treatment’, ‘pond_desc’ and ‘species’	‘SPAWN_POND’, ‘POND_STOCK’ and ‘POND_TREAT’

Load file keyword	Description	Triggering activity	Dependencies	Database tables populated
release	Loads destination and amount of aggregates released as spawn or fingerlings to external parties.	'Dissemination'	'aggregate_fam' and 'partner'	'RELEASE'
sample	Identifies tissue samples and their location.	'Tissue sampling'	'tagging'	'SAMPLE'
site	Defines sites where breeding or other activities occur.	'Nursing', 'Progeny testing' and 'Spawning'	None	'SITE'
spawn_batch	Groups spawn runs within spawn rounds into spawn batches for genetic analyses.	'Spawning'	'spawning'	'SPAWN_BATCH'
spawn_round	Identifies tissue samples and their location.	'Founders introduced', 'Nursing' and 'Spawing'	None	'SPAWN_ROUND'
spawning	Identifies families, their parents and data generated at spawning.	'Spawning'	'founder_desc' or 'tagging', and 'site'.	'SPAWN_DATE', 'FAMILY' and 'SPAWN'
outlier	Identifies outliers in measurement data.	'Data checking'	'measure_data'	'OUTLIER'
partner	Details of partners (e.g. hatcheries and nurseries).	'Dissemination'	None	'PARTNER'
pond_desc	Identifies ponds within sites.	'Nursing' and 'Progeny testing'	'site'	'POND'.
pond_treat	Describes progeny test (i.e. grow-out pond) treatments for each spawn round.	'Progeny testing'	'treatment', 'pond_desc' and 'species'	'SPAWN_POND', 'POND_STOCK' and 'POND_TREAT'
species	Enters species details.	'Founders introduced', 'Genetic analysis' and 'Progeny testing'	None	'SPECIES'
tagging	Assigns individual identifiers (and families) to fish.	'Tagging / family assignment'	'spawning'	'INDIV_PIT', 'INDIV_SPECIES' and 'INDIV_FAM'
tbl_details	Describes the tables in the GCIP Database.	'Database described'	None	'TBL_DETAILS'

Load file keyword	Description	Triggering activity	Dependencies	Database tables populated
trait	Describes the traits measured on individual tagged fish.	‘Measurement’ and ‘Genetic analysis’	None	‘TRAIT’
treatment	Describes culture, fertiliser, feed and treatment applied to ponds (e.g. progeny test ponds).	‘Genetic analysis’ and ‘Progeny testing’	‘feed’ and ‘fert’	‘TREATMENT’
species	Enters species details.	‘Founders introduced’, ‘Genetic analysis’ and ‘Progeny testing’	None	‘SPECIES’
tagging	Assigns individual identifiers (and families) to fish.	‘Tagging / family assignment’	‘spawning’	‘INDIV_PIT’, ‘INDIV_SPECIES’ and ‘INDIV_FAM’

Table 3. Load file keywords and descriptions.

2. Load order

There is a strict order in which data must be loaded to the WorldFish Carp Genetic Improvement Program Database (Figure 3).

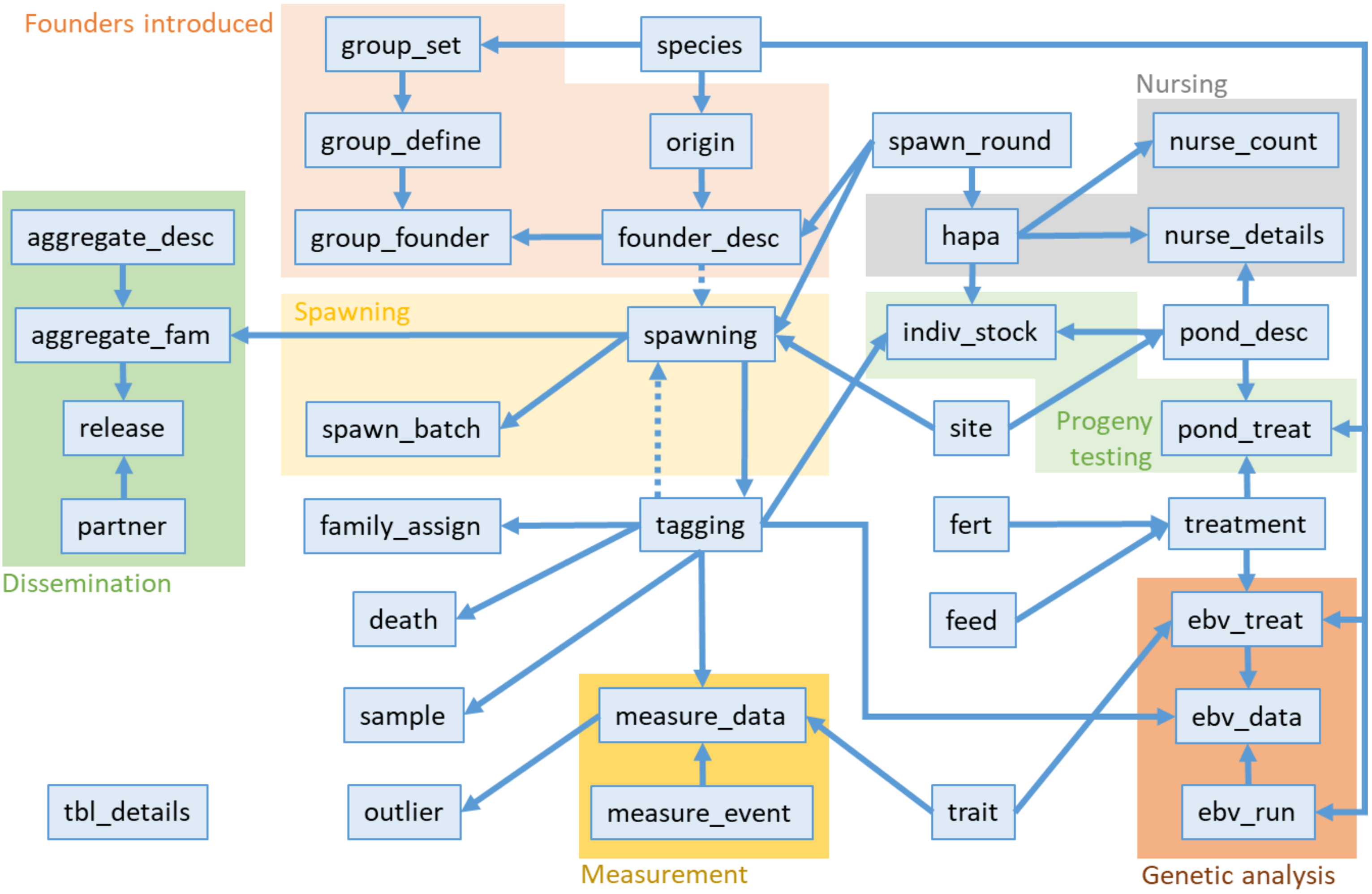


Figure 3. Load file types and keywords. The order in which load files must be loaded is indicated by arrows.

3. Description of load files by keyword

1.1. aggregate_desc

Description: Describes aggregates (Table 4; Figure 4). Aggregates are pools or groups of families.

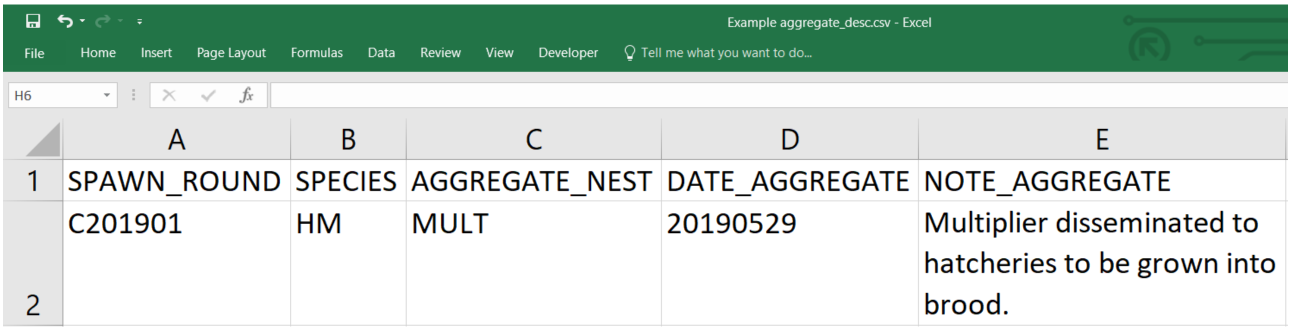
Triggering activity: 'Dissemination'.

Dependencies: 'spawning'.

Database tables populated: 'AGGREGATE_DESC'.

Column names	Description of data	Definition
SPAWN_ROUND	CHARACTER(7) NOT NULL	7 characters. First character represents the breeding program (e.g. C for carp) next four characters are the year and the final 2 characters is the spawn run within year and breeding program.
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name).
AGGREGATE_NEST	VARCHAR(15) NOT NULL	Single word descriptor of aggregate.
DATE_AGGREGATE	CHARACTER(8) NOT NULL	Date (YYYYMMDD) on which the aggregate was created (i.e. date when families were pooled).
NOTE_AGGREGATE	VARCHAR(255)	Description of the aggregate.

Table 4. Description of fields in 'aggregate_desc' load files.


Example aggregate_desc.csv - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Tell me what you want to do...
H6

	A	B	C	D	E
1	SPAWN_ROUND	SPECIES	AGGREGATE_NEST	DATE_AGGREGATE	NOTE_AGGREGATE
2	C201901	HM	MULT	20190529	Multiplier disseminated to hatcheries to be grown into brood.

Figure 4. Example of an 'aggregate_desc' load file.

1.2. aggregate_fam

Description: Assigns families to aggregates (Table 5; Figure 5).

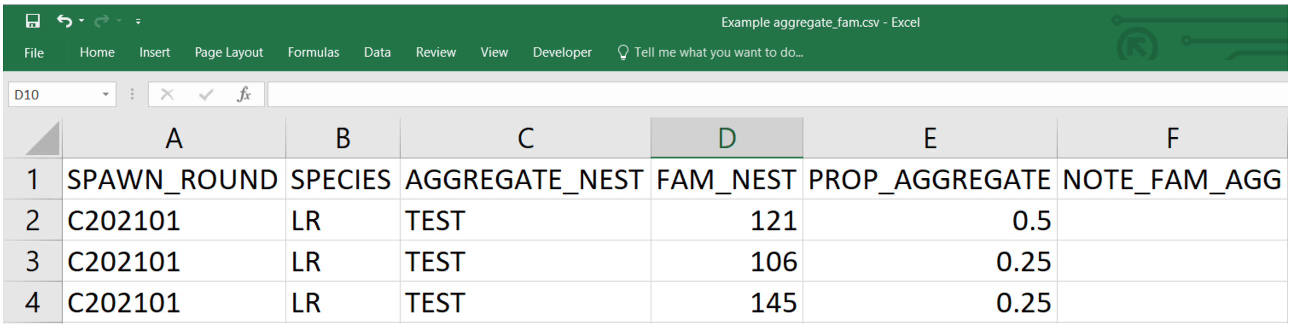
Triggering activity: 'Dissemination'.

Dependencies: 'aggregate_desc' and 'spawning'.

Database tables populated: 'AGGREGATE_FAM'.

Column names	Description of data	Definition
SPAWN_ROUND	CHARACTER(7) NOT NULL	7 characters. First character represents the breeding program (e.g. C for carp) next four characters are the year and the final 2 characters is the spawn run within year and breeding program
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
AGGREGATE_NEST	VARCHAR(15) NOT NULL	Single word descriptor of aggregate.
FAM_NEST	INTEGER(4)	Family identifier within SPAWN_ROUND and SPECIES (Integer < 10000)
PROP_AGGREGATE	REAL NOT NULL	Proportion of aggregate comprised of the family.
NOTE_FAM_AGG	VARCHAR(255)	Note on family within the aggregate.

Table 5. Description of fields in 'aggregate_fam' load files.


Example aggregate_fam.csv - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Tell me what you want to do...
D10

	A	B	C	D	E	F
1	SPAWN_ROUND	SPECIES	AGGREGATE_NEST	FAM_NEST	PROP_AGGREGATE	NOTE_FAM_AGG
2	C202101	LR	TEST	121	0.5	
3	C202101	LR	TEST	106	0.25	
4	C202101	LR	TEST	145	0.25	

Figure 5. Example of an 'aggregate_fam' load file.

1.3. death

Description: Specifies date of death for individuals (Table 6; Figure 6).

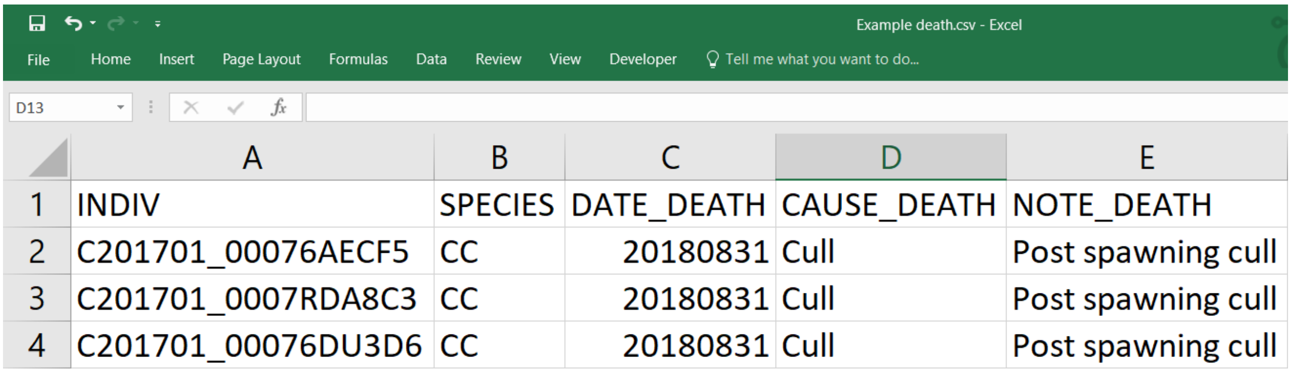
Triggering activity: ‘Death’.

Dependencies: **‘**tagging’.

Database tables populated: ‘DEATH’.

Column names	Description of data	Definition
INDIV	VARCHAR(22) NOT NULL	Up to 22 characters. The first 7 characters are the spawn round identifier, the eighth character is an underscore and the final characters are the PIT tag identifier.
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
DATE_DEATH	CHARACTER(8) NOT NULL	Date (YYYYMMDD) of spawn release/dissemination.
CAUSE_DEATH	VARCHAR(22) NOT NULL	‘Cull’, ‘Unknown’, ‘Disease_parasites’ or ‘Water_quality’.
NOTE_DEATH	VARCHAR(255)	Note on death.

Table 6. Description of fields in ‘death’ load files.



	A	B	C	D	E
1	INDIV	SPECIES	DATE_DEATH	CAUSE_DEATH	NOTE_DEATH
2	C201701_00076AECF5	CC	20180831	Cull	Post spawning cull
3	C201701_0007RDA8C3	CC	20180831	Cull	Post spawning cull
4	C201701_00076DU3D6	CC	20180831	Cull	Post spawning cull

Figure 6. Example of a ‘death’ load file.

1.4. ebv_data

Description: Loads estimated breeding values (EBVs; Table 7; Figure 7).

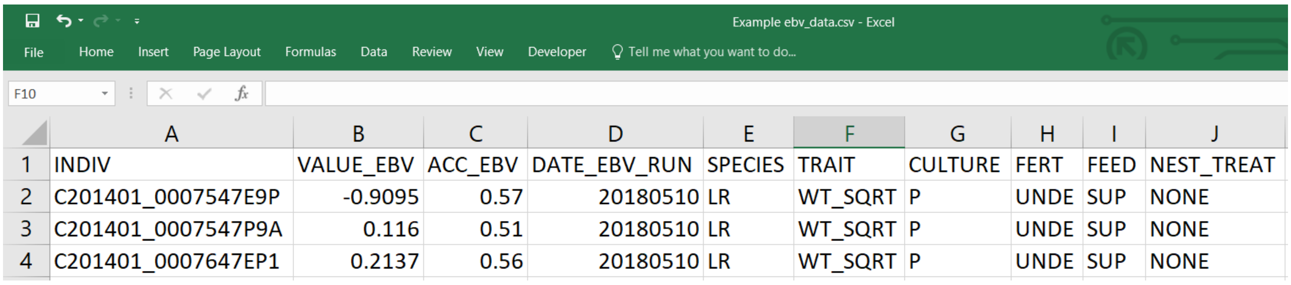
Triggering activity: ‘Genetic analysis’.

Dependencies: ‘ebv_run’, ‘ebv_treat’ and ‘tagging’.

Database tables populated: ‘EBV’.

Column names	Description of data	Definition
INDIV	VARCHAR(22) NOT NULL	Up to 22 characters. The first 7 characters are the spawn round identifier, the eighth character is an underscore and the final characters are the PIT tag identifier.
DATE_EBV_RUN	CHARACTER(8) NOT NULL	Date (YYYYMMDD) of EBV run.
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
TRAIT	VARCHAR(10) NOT NULL	TRAIT name
CULTURE	CHARACTER(1) NOT NULL	‘M’ monoculture or ‘P’ for polyculture
FERT	CHARACTER(4) NOT NULL	Fertiliser identifier
FEED	CHARACTER(3) NOT NULL	Feed identifier
NEST_TREAT	CHARACTER(4) NOT NULL	Treatment identifier
VALUE_EBV	REAL NOT NULL	EBVs in genetic standard deviation units.
ACC_EBV	REAL	EBV accuracy (0 to 1).

Table 7. Description of fields in the ‘ebv_data’ load file.



	A	B	C	D	E	F	G	H	I	J
1	INDIV	VALUE_EBV	ACC_EBV	DATE_EBV_RUN	SPECIES	TRAIT	CULTURE	FERT	FEED	NEST_TREAT
2	C201401_0007547E9P	-0.9095	0.57	20180510	LR	WT_SQRT	P	UNDE	SUP	NONE
3	C201401_0007547P9A	0.116	0.51	20180510	LR	WT_SQRT	P	UNDE	SUP	NONE
4	C201401_0007647EP1	0.2137	0.56	20180510	LR	WT_SQRT	P	UNDE	SUP	NONE

Figure 7. Example of an ‘ebv_data’ load file.

1.5. ebv_run

Description: Identifies unique estimated breeding value (EBV) run (i.e. computation; Table 8; Figure 8).

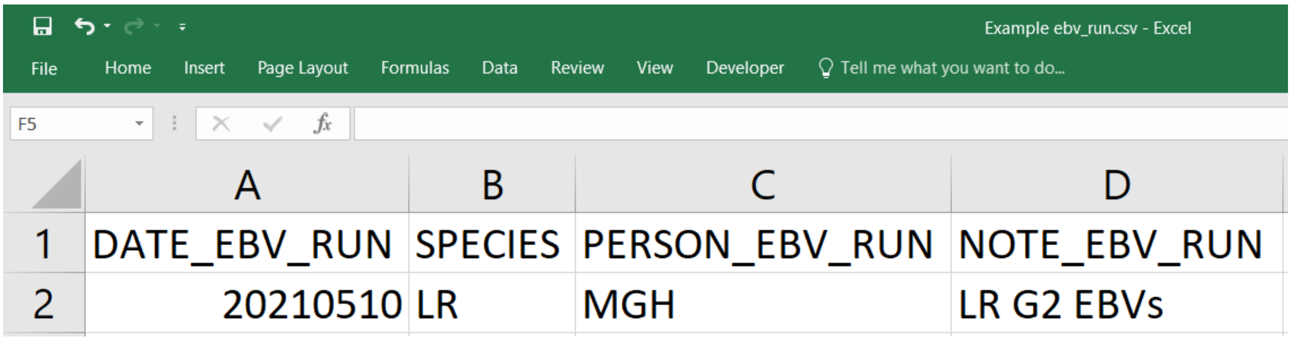
Triggering activity: 'Genetic analysis'.

Dependencies: 'species'.

Database tables populated: 'EBV_RUN'.

Column names	Description of data	Definition
DATE_EBV_RUN	CHARACTER(8) NOT NULL	Date (YYYYMMDD) of measurement.
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
PERSON_EBV_RUN	VARCHAR(50)	Person responsible for EBV run.
NOTE_EBV_RUN	VARCHAR(255)	Short note on EBV run.

Table 8. Description of fields in the 'ebv_run' load file.



	A	B	C	D
1	DATE_EBV_RUN	SPECIES	PERSON_EBV_RUN	NOTE_EBV_RUN
2	20210510	LR	MGH	LR G2 EBVs

Figure 8. Example of an 'ebv_run' load file.

1.6. ebv_treat

Description: Identifies unique combinations of trait and treatment for which estimated breeding values (EBVs) may be computed (Table 9; Figure 9).

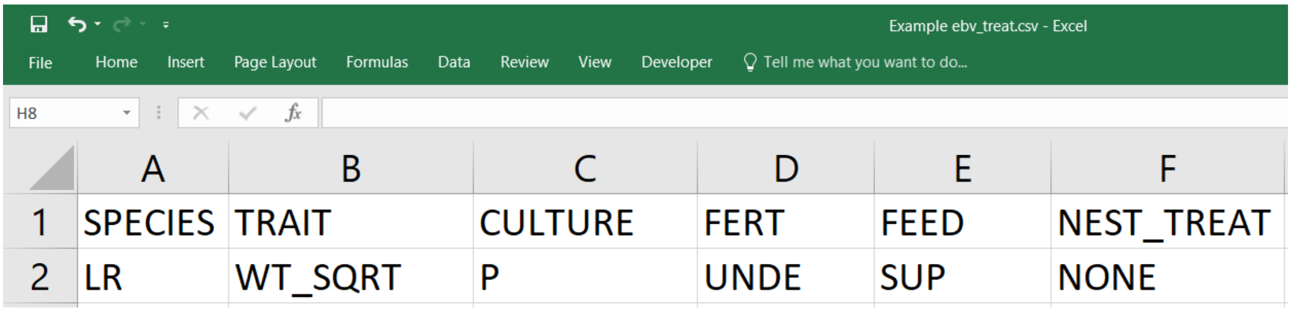
Triggering activity: ‘Genetic analysis’.

Dependencies: **‘**species’, ‘trait’, and ‘treatment’.

Database tables populated: ‘EBV_TRAIT_TREAT’.

Column names	Description of data	Definition
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
TRAIT	VARCHAR(10) NOT NULL	TRAIT name
CULTURE	CHARACTER(1) NOT NULL	‘M’ monoculture or ‘P’ for polyculture
FERT	CHARACTER(4) NOT NULL	Fertiliser identifier
FEED	CHARACTER(3) NOT NULL	Feed identifier
NEST_TREAT	CHARACTER(4) NOT NULL	Treatment identifier

Table 9. Description of fields in the ‘ebv_treat’ load file.



	A	B	C	D	E	F
1	SPECIES	TRAIT	CULTURE	FERT	FEED	NEST_TREAT
2	LR	WT_SQRT	P	UNDE	SUP	NONE

Figure 9. Example of an ‘ebv_treat’ load file.

1.7. family_assign

Description: Assigns a family to an individual if the family was unknown at the time of tagging (Table 10; Figure 10).

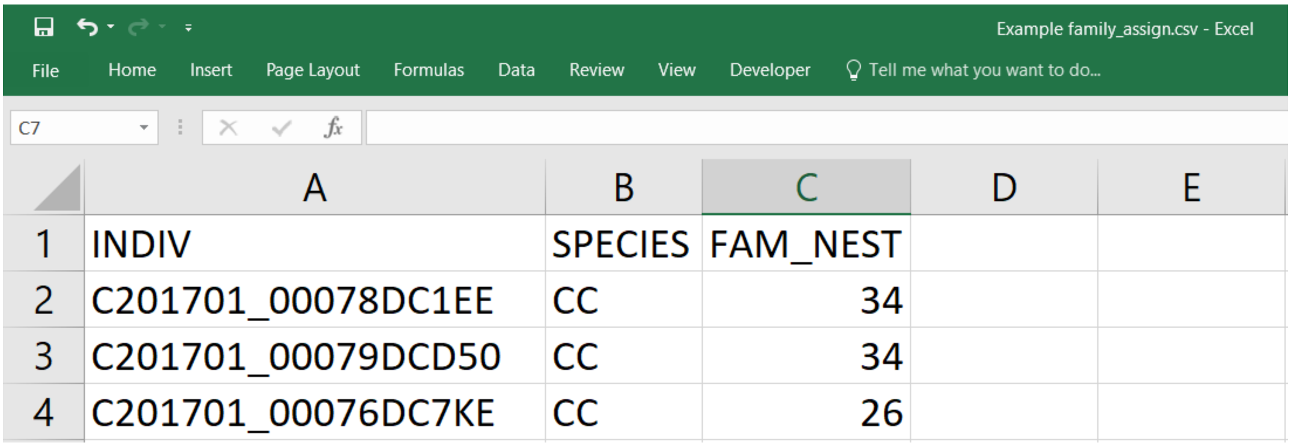
Triggering activity: ‘Tagging / family assignment’.

Dependencies: ‘tagging’.

Database tables populated: ‘INDIV_FAM’.

Column names	Description of data	Definition
INDIV	VARCHAR(22) NOT NULL	Up to 22 characters. The first 7 characters are the spawn round identifier, the eighth character is an underscore and the final characters are the PIT tag identifier.
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
FAM_NEST	INTEGER(4)	Family identifier within SPAWN_ROUND and SPECIES (Integer < 10000)

Table 10. Description of fields in ‘family_assign’ load files.


Example family_assign.csv - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Tell me what you want to do...
C7
fx

	A	B	C	D	E
1	INDIV	SPECIES	FAM_NEST		
2	C201701_00078DC1EE	CC	34		
3	C201701_00079DCD50	CC	34		
4	C201701_00076DC7KE	CC	26		

Figure 10. Example of a ‘family_assign’ load file.

1.8. feed

Description: Provides details of feeds (Table 11; Figure 11).

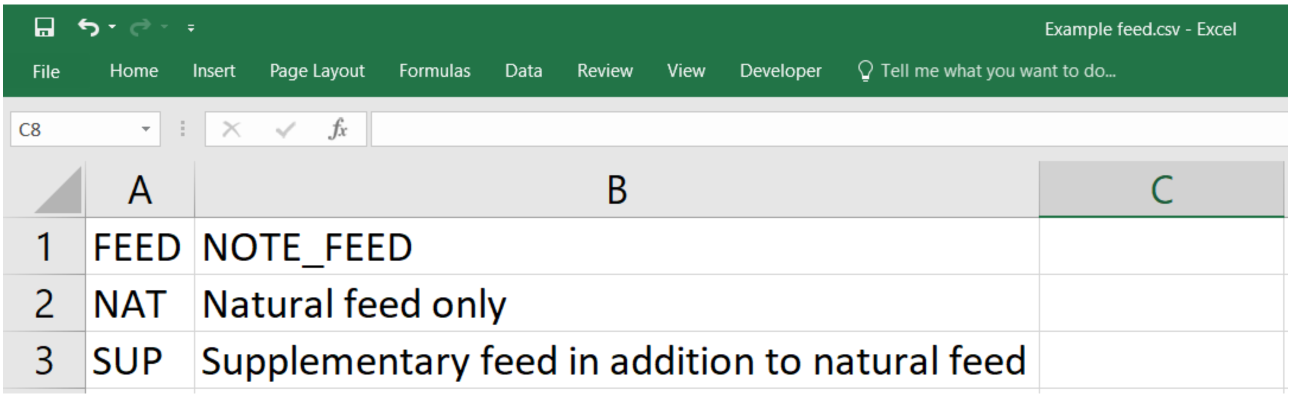
Triggering activity: ‘Genetic analysis’ and ‘Progeny testing’.

Dependencies: None.

Database tables populated: ‘FEED’.

Column names	Description of data	Definition
FEED	CHARACTER(3) NOT NULL,	Feed identifier
NOTE_FEED	VARCHAR(255)	Note on feed (e.g. composition, source, company)

Table 11. Description of fields in the ‘feed’ load file.



	A	B	C
1	FEED	NOTE_FEED	
2	NAT	Natural feed only	
3	SUP	Supplementary feed in addition to natural feed	

Figure 11. Example of a ‘feed’ load file.

1.9. fert

Description: Provides details of fertilisers (Table 12; Figure 12).

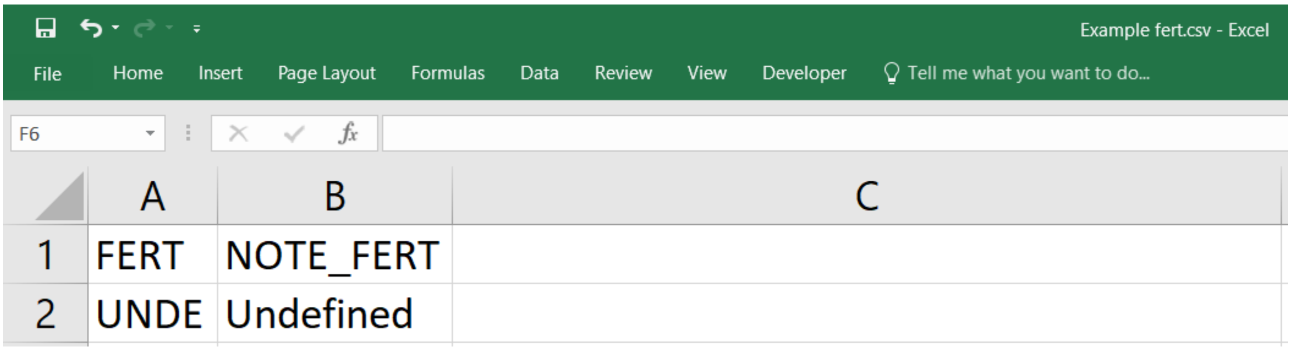
Triggering activity: ‘Genetic analysis’ and ‘Progeny testing’.

Dependencies: None.

Database tables populated: ‘FERT’.

Column names	Description of data	Definition
FERT	CHARACTER(4) NOT NULL,	Fertiliser identifier
NOTE_FERT	VARCHAR(255)	Note on fertiliser (e.g. composition, source, company)

Table 12. Description of fields in the ‘fert’ load file.



	A	B	C
1	FERT	NOTE_FERT	
2	UNDE	Undefined	

Figure 12. Example of a ‘fert’ load file.

1.10.founder_desc

Description: Assigns founders to species and origin (Table 13; Figure 13).

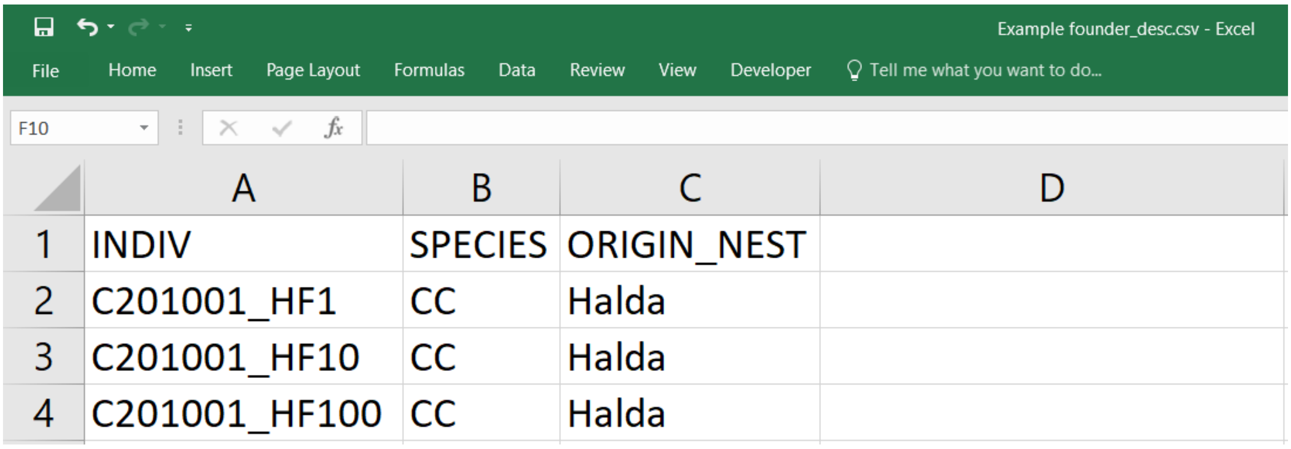
Triggering activity: 'Founders introduced'.

Dependencies: 'origin' and 'spawn_round'.

Database tables populated: 'INDIV_PIT', 'INDIV_SPECIES' and 'FOUNDER'.

Column names	Description of data	Definition
INDIV	VARCHAR (22) NOT NULL	Up to 22 characters. The first 7 characters are the spawn round identifier, the eighth character is an underscore and the final characters are the PIT tag identifier.
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
ORIGIN_NEST	VARCHAR(10) NOT NULL	Single word descriptor of origin within SPECIES.

Table 13. Description of fields in the 'founder_desc' load file.


Example founder_desc.csv - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Tell me what you want to do...
F10

	A	B	C	D
1	INDIV	SPECIES	ORIGIN_NEST	
2	C201001_HF1	CC	Halda	
3	C201001_HF10	CC	Halda	
4	C201001_HF100	CC	Halda	

Figure 13. Example of a 'founder_desc' load file.

1.11.group_define

Description: Defines genetic groups within a previously defined ‘group set’ (Table 14; Figure 14).

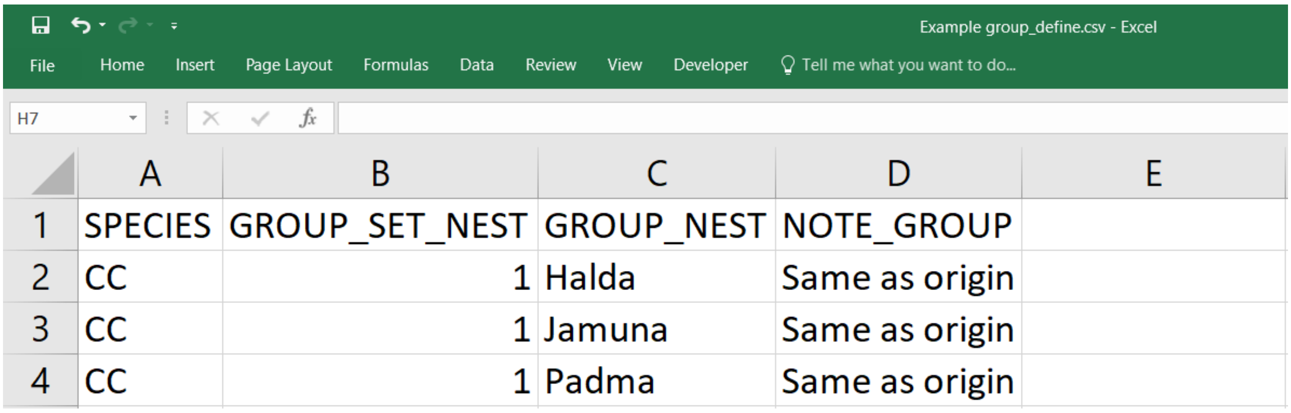
Triggering activity: ‘Founders introduced’.

Dependencies: ‘group_set’.

Database tables populated: ‘GROUP_DEFINE’.

Column names	Description of data	Definition
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
GROUP_SET_NEST	INTEGER NOT NULL	Group set identifier within species (integer)
GROUP_NEST	VARCHAR(20) NOT NULL	Name of group within GROUP_SET
NOTE_GROUP	VARCHAR(255)	Description of genetic group within SPECIES and GROUP_SET_NEST

Table 14. Description of fields in the ‘group_define’ load file.


Example group_define.csv - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Tell me what you want to do...
H7
fx

	A	B	C	D	E
1	SPECIES	GROUP_SET_NEST	GROUP_NEST	NOTE_GROUP	
2	CC	1	Halda	Same as origin	
3	CC	1	Jamuna	Same as origin	
4	CC	1	Padma	Same as origin	

Figure 14. Example of a ‘group_define’ load file.

1.12.group_founder

Description: Assigns founders to genetic groups (Table 15; Figure 15).

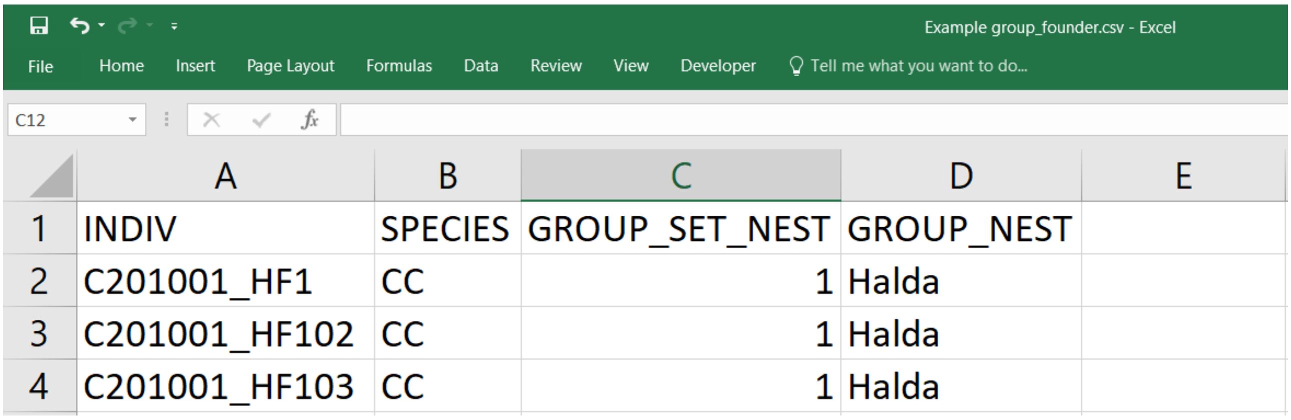
Triggering activity: ‘Founders introduced’.

Dependencies: ‘group_define’ and ‘founder_desc’.

Database tables populated: ‘GROUP_FOUNDER’.

Column names	Description of data	Definition
INDIV	VARCHAR(22) NOT NULL	Up to 22 characters. The first 7 characters are the spawn round identifier, the eighth character is an underscore and the final characters are the PIT tag identifier.
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
GROUP_SET_NEST	INTEGER NOT NULL	Group set identifier within species (integer)
GROUP_NEST	VARCHAR(20) NOT NULL	Name of group within GROUP_SET

Table 15. Description of fields in the ‘group_founder’ load file.


Example group_founder.csv - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Tell me what you want to do...
C12

	A	B	C	D	E
1	INDIV	SPECIES	GROUP_SET_NEST	GROUP_NEST	
2	C201001_HF1	CC	1	Halda	
3	C201001_HF102	CC	1	Halda	
4	C201001_HF103	CC	1	Halda	

Figure 15. Example of a ‘group_founder’ load file.

1.13.group_set

Description: Defines a 'set' of genetic groups (Table 16; Figure 16). Each founder can be assigned to a genetic group in multiple genetic group 'sets'.

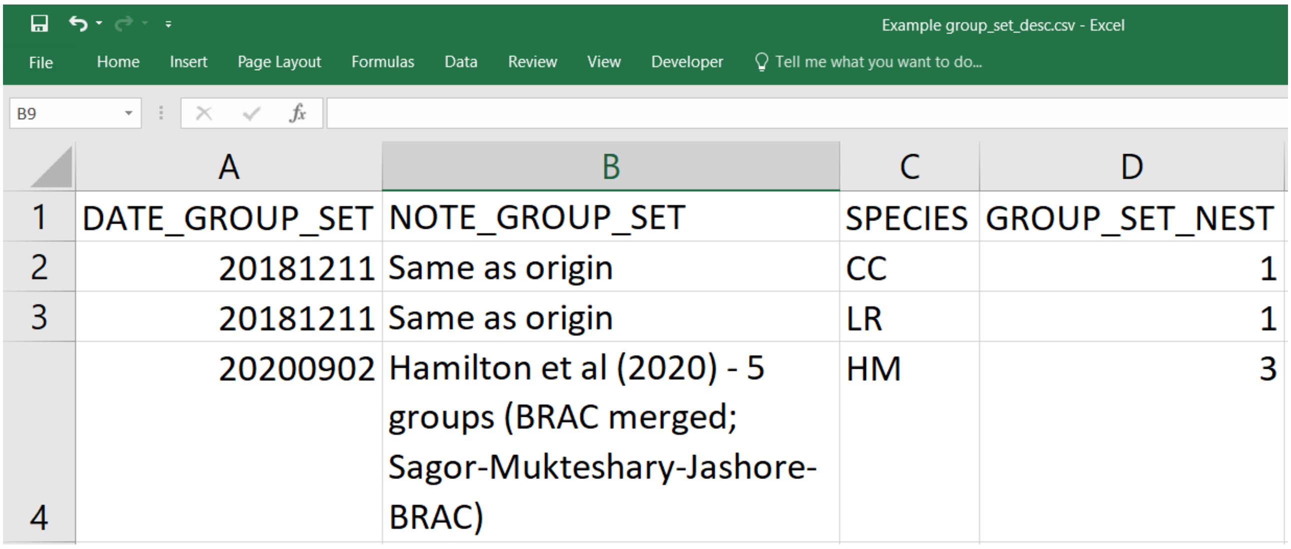
Triggering activity: 'Founders introduced'.

Dependencies: 'species'.

Database tables populated: 'GROUP_SET_DESC'.

Column names	Description of data	Definition
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
GROUP_SET_NEST	INTEGER NOT NULL	Group set identifier within species (integer)
DATE_GROUP_SET	CHARACTER(8) NOT NULL	Date (YYYYMMDD) on which set of genetic groups was defined
NOTE_GROUP_SET	VARCHAR(255) NOT NULL	Description of set of genetic groups

Table 16. Description of fields in the 'group_set' load file.


Example group_set_desc.csv - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Tell me what you want to do...
B9

	A	B	C	D
1	DATE_GROUP_SET	NOTE_GROUP_SET	SPECIES	GROUP_SET_NEST
2	20181211	Same as origin	CC	1
3	20181211	Same as origin	LR	1
4	20200902	Hamilton et al (2020) - 5 groups (BRAC merged; Sagor-Mukteshary-Jashore-BRAC)	HM	3

Figure 16. Example of a 'group_set' load file.

1.14.hapa

Description: Defines a nursing hapa (or chamber/pond) identifier within a spawn round (Table 17; Figure 17).

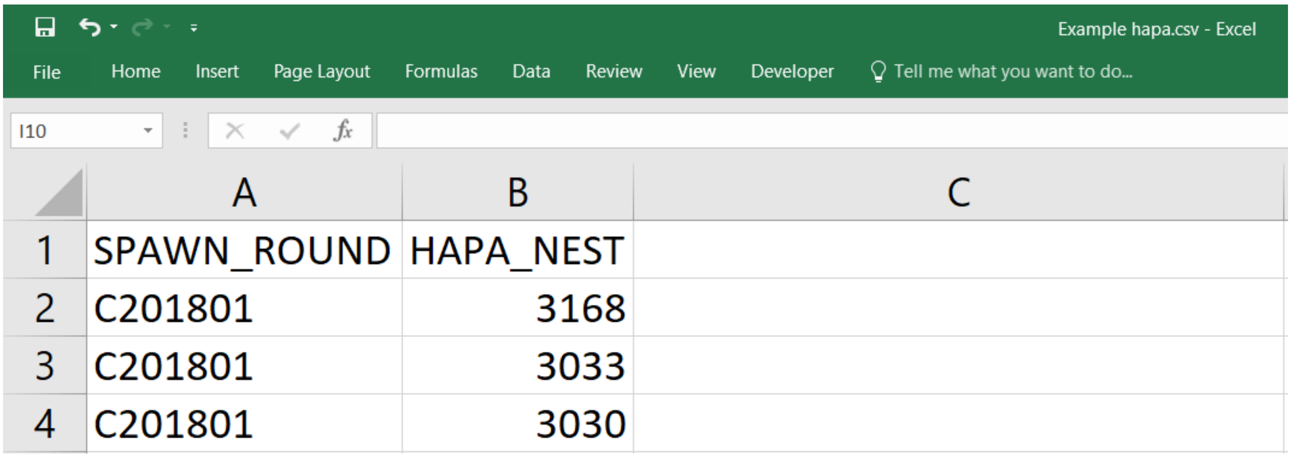
Triggering activity: 'Nursing'.

Dependencies: 'spawn_round'.

Database tables populated: 'HAPA'.

Column names	Description of data	Definition
SPAWN_ROUND	CHARACTER(7) NOT NULL	7 characters. First character represents the breeding program (e.g. C for carp) next four characters are the year and the final 2 characters is the spawn run within year and breeding program
HAPA_NEST	INTEGER NOT NULL	Nurse hapa within SPAWN_ROUND (e.g. hapa/chamber identifier used to define nurse hapa throughout nursing; even if moved from one chamber to another - may be derived from FAM)

Table 17. Description of fields in the 'hapa' load file.



	A	B	C
1	SPAWN_ROUND	HAPA_NEST	
2	C201801	3168	
3	C201801	3033	
4	C201801	3030	

Figure 17. Example of a 'hapa' load file.

1.15.indiv_stock

Description: Assigns individuals to a grow-out pond (Table 18; Figure 18).

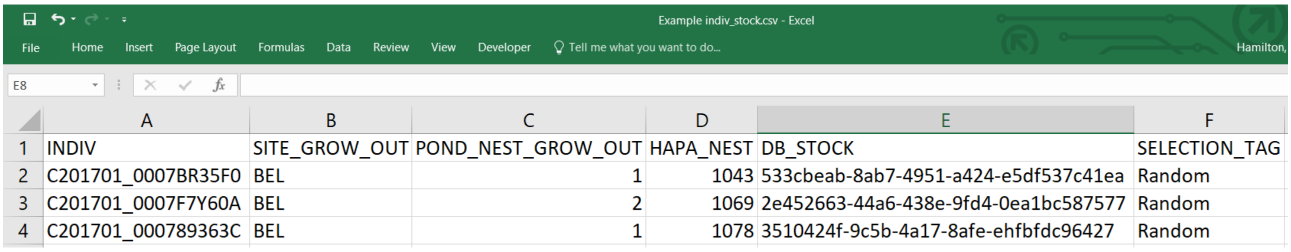
Triggering activity: ‘Progeny testing’.

Dependencies: ‘hapa’, ‘pond_desc’ and ‘tagging’.

Database tables populated: ‘INDIV_STOCK’.

Column names	Description of data	Definition
INDIV	VARCHAR(22) NOT NULL	Up to 22 characters. The first 7 characters are the spawn round identifier, the eighth character is an underscore and the final characters are the PIT tag identifier.
SITE_GROW_OUT	CHARACTER(3) NOT NULL	Three character site identifier
POND_NEST_GROW_OUT	INTEGER NOT NULL	Pond identifier within site (integer < 1000)
HAPA_NEST	INTEGER NOT NULL	Nurse hapa within SPAWN_ROUND (e.g. hapa/chamber identifier used to define nurse hapa throughout nursing; even if moved from one chamber to another - may be derived from FAM)
DB_STOCK	CHARACTER(36)	Identifier in WorldFish data capture system Stocking table
SELECTION_TAG	CHARACTER(6) NOT NULL	Selection undertaken at tagging (‘Random’ or ‘Best’)

Table 18. Description of fields in the ‘indiv_stock’ load file.



	A	B	C	D	E	F
1	INDIV	SITE_GROW_OUT	POND_NEST_GROW_OUT	HAPA_NEST	DB_STOCK	SELECTION_TAG
2	C201701_0007BR35F0	BEL	1	1043	533cbeab-8ab7-4951-a424-e5df537c41ea	Random
3	C201701_0007F7Y60A	BEL	2	1069	2e452663-44a6-438e-9fd4-0ea1bc587577	Random
4	C201701_000789363C	BEL	1	1078	3510424f-9c5b-4a17-8afe-ehfbfdc96427	Random

Figure 18. Example of an ‘indiv_stock’ load file.

1.16.measure_data

Description: Loads measurement data (Table 19; Figure 19).

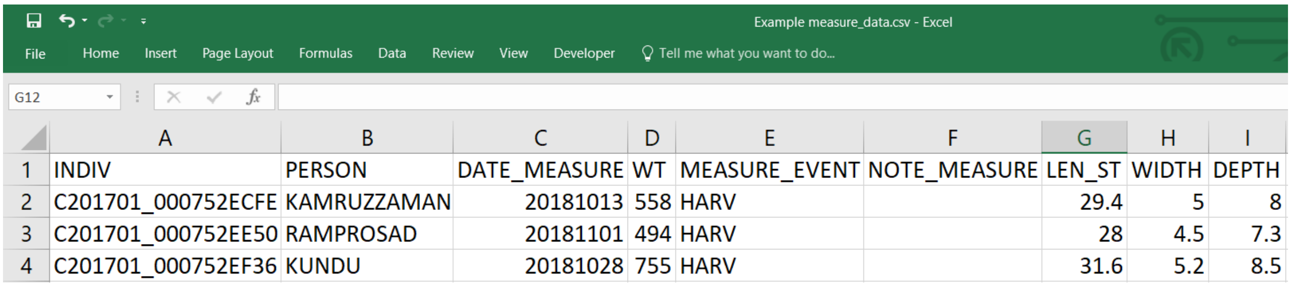
Triggering activity: ‘Progeny testing’.

Dependencies: ‘measure_event’, ‘tagging’, ‘trait’.

Database tables populated: ‘MEASURE_DETAILS’ and ‘MEASURE’.

Column names	Description of data	Definition
INDIV	VARCHAR(22) NOT NULL	Up to 22 characters. The first 7 characters are the spawn round identifier, the eighth character is an underscore and the final characters are the PIT tag identifier.
MEASURE_EVENT	VARCHAR(20) NOT NULL	Measurement event identifier
Data for TRAIT 1	REAL NOT NULL	Column heading is the TRAIT identifier (see ‘trait’ load file. The values in the column are the measured values for the trait.
…		Data for multiple traits can be loaded.
PERSON	VARCHAR(50)	Person responsible for the measurement
NOTE_MEASURE	VARCHAR(255)	Short note on individual measurement
DATE_MEASURE	CHARACTER(8) NOT NULL	Date (YYYYMMDD) of measurement

Table 19. Description of fields in the ‘measure_data’ load file.



Example measure_data.csv - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Tell me what you want to do...
G12

	A	B	C	D	E	F	G	H	I
1	INDIV	PERSON	DATE_MEASURE	WT	MEASURE_EVENT	NOTE_MEASURE	LEN_ST	WIDTH	DEPTH
2	C201701_000752ECFE	KAMRUZZAMAN	20181013	558	HARV		29.4	5	8
3	C201701_000752EE50	RAMPROSAD	20181101	494	HARV		28	4.5	7.3
4	C201701_000752EF36	KUNDU	20181028	755	HARV		31.6	5.2	8.5

Figure 19. Example of a ‘measure_data’ load file.

1.17.measure_event

Description: Defines measurement 'events'. All measurement data must ultimately be assigned to a measurement event (Table 20; Figure 20).

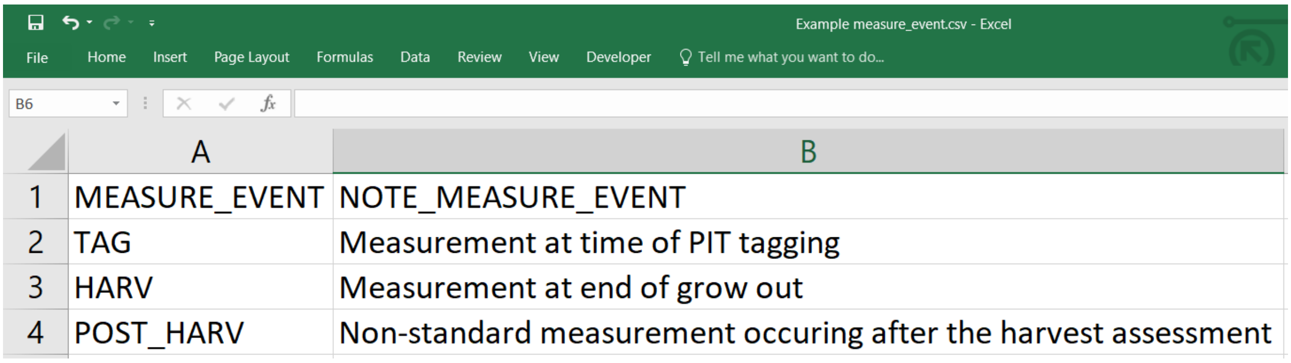
Triggering activity: 'Progeny testing'.

Dependencies: None.

Database tables populated: 'MEASURE_EVENT'.

Column names	Description of data	Definition
MEASURE_EVENT	VARCHAR(20) NOT NULL	Measurement event identifier
NOTE_MEASURE_EVENT	VARCHAR(255) NOT NULL	Details of measurement event

Table 20. Description of fields in the 'measure_event' load file.


Example measure_event.csv - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Tell me what you want to do...
B6
fx

	A	B
1	MEASURE_EVENT	NOTE_MEASURE_EVENT
2	TAG	Measurement at time of PIT tagging
3	HARV	Measurement at end of grow out
4	POST_HARV	Non-standard measurement occuring after the harvest assessment

Figure 20. Example of a 'measure_event' load file.

1.18.nurse_count

Description: Enters the count of individuals surviving in each hapa at the time hapa transfer and tagging (Table 21; Figure 21).

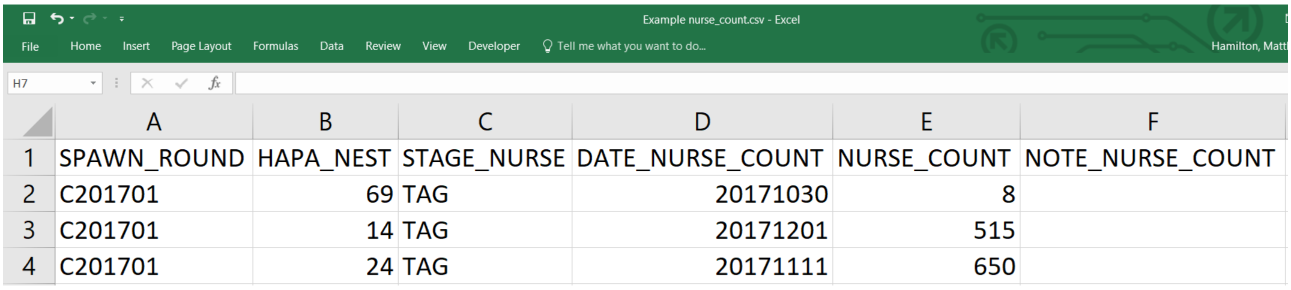
Triggering activity: ‘Nursing’.

Dependencies: ‘hapa’.

Database tables populated: ‘NURSE_COUNT’.

Column names	Description of data	Definition
SPAWN_ROUND	CHARACTER(7) NOT NULL	7 characters. First character represents the breeding program (e.g. C for carp) next four characters are the year and the final 2 characters is the spawn run within year and breeding program
HAPA_NEST	INTEGER NOT NULL	Nurse hapa within SPAWN_ROUND (e.g. hapa/chamber identifier used to define nurse hapa throughout nursing; even if moved from one chamber to another - may be derived from FAM)
STAGE_NURSE	VARCHAR(5) NOT NULL	Nursing stage starting - 'BLUE' when moving from white hapas, 'BLACK' when moving from blue hapas and 'TAG' when tagging from black hapas.
DATE_NURSE_COUNT	CHARACTER(8) NOT NULL	Date hapa or chamber stocked (YYYYMMDD)
NURSE_COUNT	INTEGER NOT NULL	Count of individuals from hapa at time of hapa transfer or tagging
NOTE_NURSE_COUNT	VARCHAR(255)	Note on NURSE_COUNT

Table 21. Description of fields in the ‘nurse_count’ load file.



	A	B	C	D	E	F
1	SPAWN_ROUND	HAPA_NEST	STAGE_NURSE	DATE_NURSE_COUNT	NURSE_COUNT	NOTE_NURSE_COUNT
2	C201701	69	TAG	20171030	8	
3	C201701	14	TAG	20171201	515	
4	C201701	24	TAG	20171111	650	

Figure 21. Example of a ‘nurse_count’ load file.

1.19.nurse_details

Description: Enters details of hapa positions, mesh size and movements during nursing. Note that a hapa identifier can move positions in a pond/s (Table 22; Figure 22).

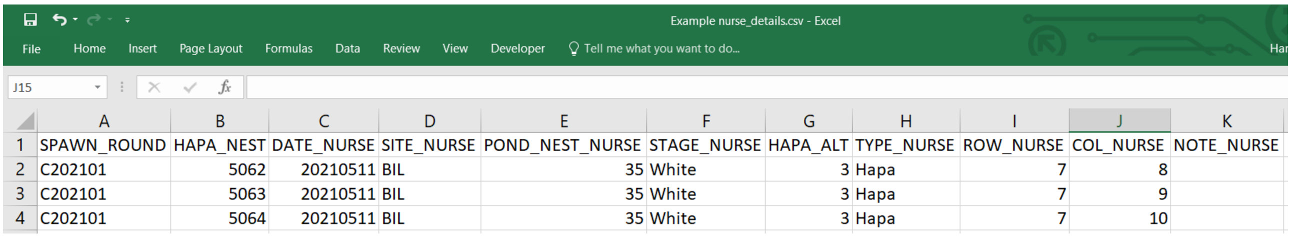
Triggering activity: ‘Nursing’.

Dependencies: ‘hapa’ and ‘pond_desc’.

Database tables populated: ‘NURSE_DETAILS’.

Column names	Description of data	Definition
SPAWN_ROUND	CHARACTER(7) NOT NULL	7 characters. First character represents the breeding program (e.g. C for carp) next four characters are the year and the final 2 characters is the spawn run within year and breeding program
HAPA_NEST	INTEGER NOT NULL	Nurse hapa within SPAWN_ROUND (e.g. hapa/chamber identifier used to define nurse hapa throughout nursing; even if moved from one chamber to another - may be derived from FAM)
DATE_NURSE	CHARACTER(8) NOT NULL	Date (YYYYMMDD) hapa or chamber stocked
SITE_NURSE	CHARACTER(3) NOT NULL	Three character site identifier
POND_NEST_NURSE	INTEGER NOT NULL	Pond identifier within site (integer < 1000)
STAGE_NURSE	VARCHAR(5)	Nursing stage (Hapa colour – ‘WHITE’, ‘BLUE’ or ‘BLACK’). Generally ‘White’ unless hapa positions are changed.
HAPA_ALT	INTEGER	Hapa/chamber identifier used for a particular stage of nursing - may be the same as HAPA_NEST
TYPE_NURSE	VARCHAR(7) NOT NULL	Chamber' or 'Hapa'
ROW_NURSE	INTEGER NOT NULL	Row within first nursery pond (rows run approximately North-South with row 1 the most northerly)
COL_NURSE	INTEGER NOT NULL	Column within first nursery pond (columns run approximately West-East with column 1 the most Westerly)
NOTE_NURSE	VARCHAR(255)	Note on NURSE

Table 22. Description of fields in the ‘nurse_details’ load file.


Example nurse_details.csv - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Tell me what you want to do...
J15

	A	B	C	D	E	F	G	H	I	J	K
1	SPAWN_ROUND	HAPA_NEST	DATE_NURSE	SITE_NURSE	POND_NEST_NURSE	STAGE_NURSE	HAPA_ALT	TYPE_NURSE	ROW_NURSE	COL_NURSE	NOTE_NURSE
2	C202101	5062	20210511	BIL	35	White	3	Hapa	7	8	
3	C202101	5063	20210511	BIL	35	White	3	Hapa	7	9	
4	C202101	5064	20210511	BIL	35	White	3	Hapa	7	10	

Figure 22. Example of a ‘nurse_details’ load file.

1.20.origin

Description: Defines origins of founders (e.g. river and coordinates, hatchery). All founders must be assigned to an origin (Table 23; Figure 23).

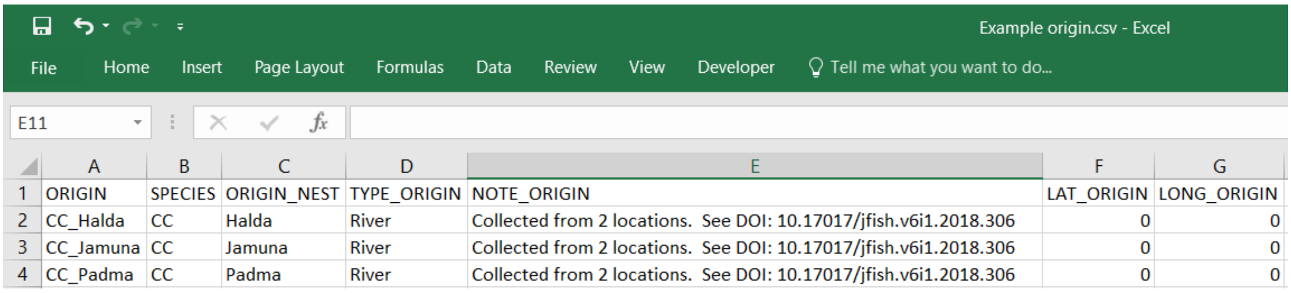
Triggering activity: ‘Founders introduced’.

Dependencies: ‘species’.

Database tables populated: ‘ORIGIN’.

Column names	Description of data	Definition
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
ORIGIN_NEST	VARCHAR(10) NOT NULL	Single word descriptor of origin within SPECIES
TYPE_ORIGIN	VARCHAR(8) NOT NULL	Either 'River' or 'Hatchery'
NOTE_ORIGIN	VARCHAR(255) NOT NULL	Description of origin
LAT_ORIGIN	REAL NOT NULL	Latitude in decimal degrees
LONG_ORIGIN	REAL NOT NULL	Longitude in decimal degrees

Table 23. Description of fields in the ‘origin’ load file.



	A	B	C	D	E	F	G
1	ORIGIN	SPECIES	ORIGIN_NEST	TYPE_ORIGIN	NOTE_ORIGIN	LAT_ORIGIN	LONG_ORIGIN
2	CC_Halda	CC	Halda	River	Collected from 2 locations. See DOI: 10.17017/jfish.v6i1.2018.306	0	0
3	CC_Jamuna	CC	Jamuna	River	Collected from 2 locations. See DOI: 10.17017/jfish.v6i1.2018.306	0	0
4	CC_Padma	CC	Padma	River	Collected from 2 locations. See DOI: 10.17017/jfish.v6i1.2018.306	0	0

Figure 23. Example of an ‘origin’ load file.

1.21.outlier

Description: Identifies outliers in measurement data (Table 24; Figure 24).

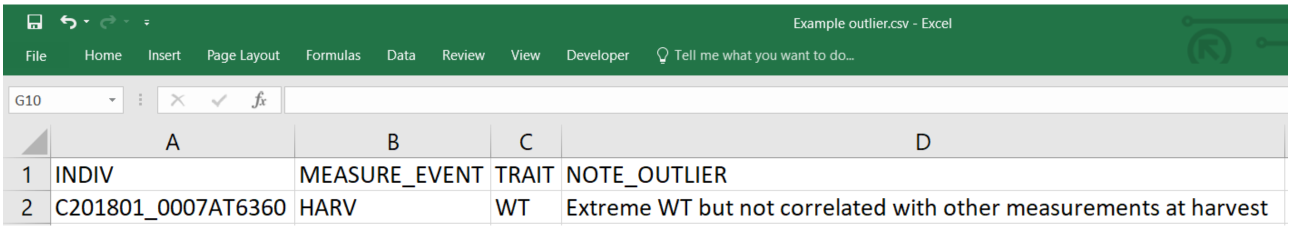
Triggering activity: ‘Data checking’.

Dependencies: **‘**measure_data’.

Database tables populated: ‘OUTLIER’.

Column names	Description of data	Definition
INDIV	VARCHAR(22) NOT NULL	Up to 22 characters. The first 7 characters are the spawn round identifier, the eighth character is an underscore and the final characters are the PIT tag identifier.
MEASURE_EVENT	VARCHAR(20) NOT NULL	Measurement event identifier
TRAIT	VARCHAR(10) NOT NULL	TRAIT name
NOTE_OUTLIER	VARCHAR(255) NOT NULL	Note on why observation is an outlier

Table 24. Description of fields in the ‘outlier’ load file.



	A	B	C	D
1	INDIV	MEASURE_EVENT	TRAIT	NOTE_OUTLIER
2	C201801_0007AT6360	HARV	WT	Extreme WT but not correlated with other measurements at harvest

Figure 24. Example of an ‘outlier’ load file.

1.22.partner

Description: Details of partners (e.g. hatcheries and nurseries) (Table 25 Figure 25).

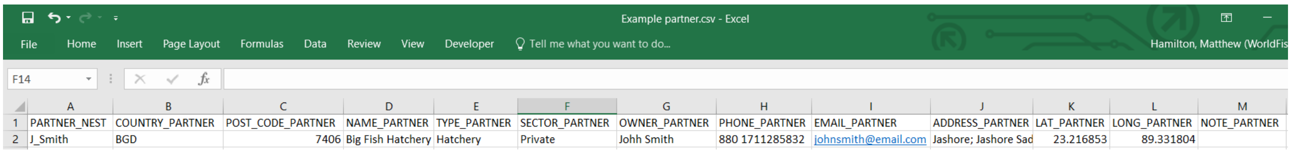
Triggering activity: ‘Dissemination’.

Dependencies: None.

Database tables populated: ‘PARTNER’.

Column names	Description of data	Definition
PARTNER_NEST	VARCHAR(15) NOT NULL	Short partner name.
COUNTRY_PARTNER	CHARACTER(3) NOT NULL	3-letter country code (ISO-3166 alpha3) of partner hatchery or nursery (e.g. BGD for Bangladesh).
POST_CODE_PARTNER	INTEGER NOT NULL	Post/zip code of partner hatchery or nursery (6 digits or less)
NAME_PARTNER	VARCHAR(255) NOT NULL	Full name of partner hatchery or nursery
TYPE_PARTNER	VARCHAR(16) NOT NULL	‘Hatchery’, ‘Nursery’ or ‘Hatchery-Nursery’
SECTOR_PARTNER	VARCHAR(7) NOT NULL	‘Private’, ‘Public’ or ‘NGO’
OWNER_PARTNER	VARCHAR(255) NOT NULL	Name of owner of hatchery or nursery
PHONE_PARTNER	VARCHAR(15) NOT NULL	Contact phone number
EMAIL_PARTNER	VARCHAR(255) NOT NULL	Contact email address
ADDRESS_PARTNER	VARCHAR(255) NOT NULL	Postal address of partner hatchery or nursery
LAT_PARTNER	REAL NOT NULL	Latitude of partner hatchery or nursery
LONG_PARTNER	REAL NOT NULL	Longitude of partner hatchery or nursery
NOTE_PARTNER	VARCHAR(255)	Note on partner hatchery or nursery

Table 25. Description of fields in the ‘partner’ load file.


Example partner.csv - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Tell me what you want to do...
Hamilton, Matthew (WorldFis
F14

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	PARTNER_NEST	COUNTRY_PARTNER	POST_CODE_PARTNER	NAME_PARTNER	TYPE_PARTNER	SECTOR_PARTNER	OWNER_PARTNER	PHONE_PARTNER	EMAIL_PARTNER	ADDRESS_PARTNER	LAT_PARTNER	LONG_PARTNER	NOTE_PARTNER
2	J_Smith	BGD	7406	Big Fish Hatchery	Hatchery	Private	Johh Smith	880 1711285832	johnsmith@email.com	Jashore; Jashore Sad	23.216853	89.331804	

Figure 25. Example of a ‘partner’ load file.

1.23.pond_desc

Description: Identifies ponds within sites (Table 26; Figure 26).

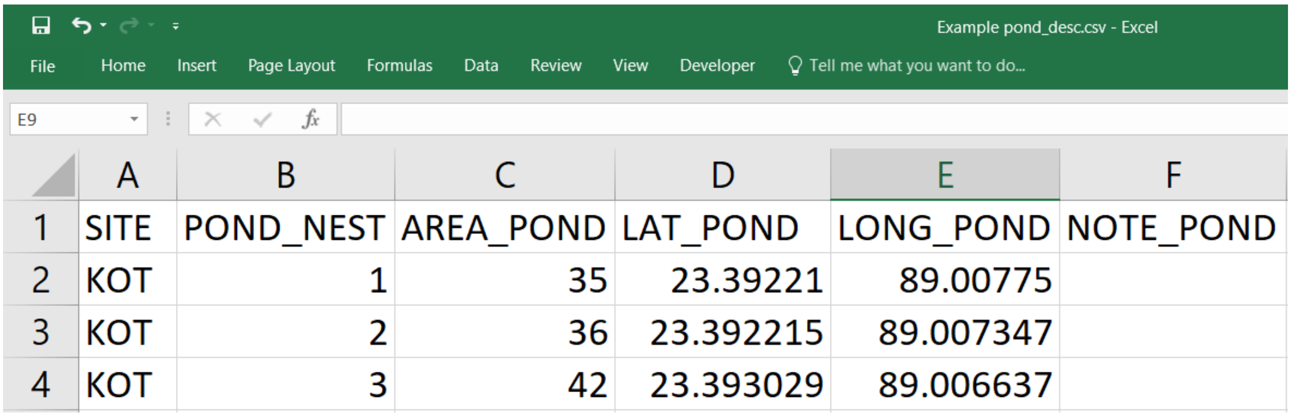
Triggering activity: ‘Nursing’ and ‘Progeny testing’.

Dependencies: **‘**site’.

Database tables populated: ‘POND’.

Column names	Description of data	Definition
SITE	CHARACTER(3) NOT NULL	Three character site identifier
POND_NEST	INTEGER NOT NULL	Pond identifier within site (integer < 1000)
AREA_POND	REAL NOT NULL	Area of pond in ha
LAT_POND	REAL NOT NULL	Latitude in decimal degrees if centre of pond
LONG_POND	REAL NOT NULL	Longitude in decimal degrees if centre of pond
NOTE_POND	VARCHAR(255)	Any notable features of the pond

Table 26. Description of fields in the ‘pond_desc’ load file.



	A	B	C	D	E	F
1	SITE	POND_NEST	AREA_POND	LAT_POND	LONG_POND	NOTE_POND
2	KOT	1	35	23.39221	89.00775	
3	KOT	2	36	23.392215	89.007347	
4	KOT	3	42	23.393029	89.006637	

Figure 26. Example of a ‘pond_desc’ load file.

1.24.pond_treat

Description: Describes progeny test (i.e. grow-out pond) treatments for each spawn round (Table 27; Figure 27).

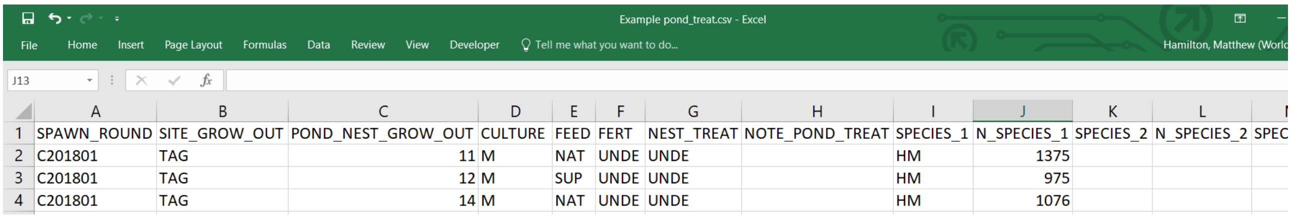
Triggering activity: ‘Progeny testing’.

Dependencies: ‘treatment’, ‘pond_desc’, and ‘species’.

Database tables populated: ‘SPAWN_POND’, ‘POND_STOCK’ and ‘POND_TREAT’.

Column names	Description of data	Definition
SPAWN_ROUND	CHARACTER(7) NOT NULL	7 characters. First character represents the breeding program (e.g. C for carp) next four characters are the year and the final 2 characters is the spawn run within year and breeding program
SITE_GROW_OUT	CHARACTER(3) NOT NULL	Three character site identifier
POND_NEST_GROW_OUT	INTEGER NOT NULL	Pond identifier within site (integer < 1000)
CULTURE	CHARACTER(1) NOT NULL	[Derived from TREAT] Culture used to grow out fish (Mono or Poly)
FEED	CHARACTER(3) NOT NULL	Feed identifier
FERT	CHARACTER(4) NOT NULL	Fertiliser identifier
NEST_TREAT		
NOTE_POND_TREAT		
SPECIES_1; SPECIES_2; SPECIES_3; SPECIES_4; SPECIES_5	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
N_SPECIES_1; N_SPECIES_2; N_SPECIES_3; N_SPECIES_4; N_SPECIES_5	INTEGER	Number of individuals stocked from the species in SPECIES

Table 27. Description of fields in the ‘trait’ load file.



	A	B	C	D	E	F	G	H	I	J	K	L	
1	SPAWN_ROUND	SITE_GROW_OUT	POND_NEST_GROW_OUT	CULTURE	FEED	FERT	NEST_TREAT	NOTE_POND_TREAT	SPECIES_1	N_SPECIES_1	SPECIES_2	N_SPECIES_2	SPEC
2	C201801	TAG	11	M	NAT	UNDE	UNDE		HM	1375			
3	C201801	TAG	12	M	SUP	UNDE	UNDE		HM	975			
4	C201801	TAG	14	M	NAT	UNDE	UNDE		HM	1076			

Figure 27. Example of a ‘trait’ load file.

1.25.release

Description: Loads destination and amount of aggregates released as spawn or fingerlings to external parties (Table 28; Figure 28).

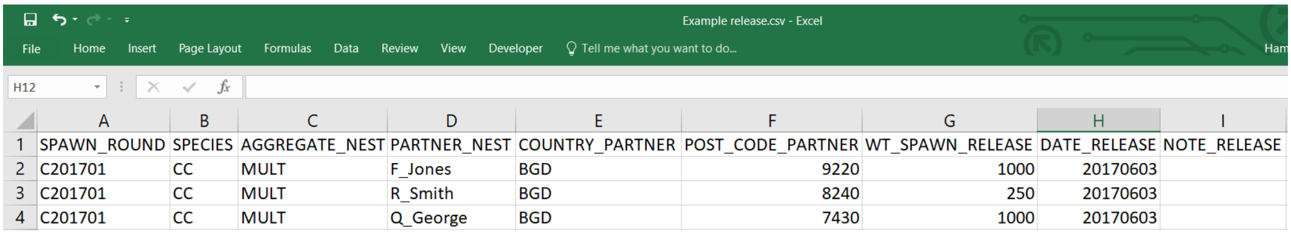
Triggering activity: ‘Dissemination’.

Dependencies: ‘aggregate_fam’ and ‘partner’.

Database tables populated: ‘RELEASE’.

Column names	Description of data	Definition
SPAWN_ROUND	CHARACTER(7) NOT NULL	7 characters. First character represents the breeding program (e.g. C for carp) next four characters are the year and the final 2 characters is the spawn run within year and breeding program.
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name).
AGGREGATE_NEST	VARCHAR(15) NOT NULL	Single word descriptor of aggregate.
PARTNER_NEST	VARCHAR(15) NOT NULL	A short partner name.
COUNTRY_PARTNER	CHARACTER(3) NOT NULL	3-letter country code (ISO-3166 alpha3) of partner hatchery or nursery (e.g. BGD for Bangladesh).
POST_CODE_PARTNER	INTEGER NOT NULL	Post/zip code of partner hatchery or nursery (6 digits or less).
WT_SPAWN_RELEASE or COUNT_FISH_RELEASE	INTEGER NOT NULL	Grams of spawn released or ‘count of fish released.
DATE_RELEASE	CHARACTER(8) NOT NULL	Date (YYYYMMDD) of spawn release/dissemination.
NOTE_RELEASE	VARCHAR(255)	Note on spawn release.

Table 28. Description of fields in ‘release’ load files.



	A	B	C	D	E	F	G	H	I
1	SPAWN_ROUND	SPECIES	AGGREGATE_NEST	PARTNER_NEST	COUNTRY_PARTNER	POST_CODE_PARTNER	WT_SPAWN_RELEASE	DATE_RELEASE	NOTE_RELEASE
2	C201701	CC	MULT	F_Jones	BGD	9220	1000	20170603	
3	C201701	CC	MULT	R_Smith	BGD	8240	250	20170603	
4	C201701	CC	MULT	Q_George	BGD	7430	1000	20170603	

Figure 28. Example of a ‘release’ load file.

1.26.sample

Description: Identifies tissue samples and their location (Table 29; Figure 29).

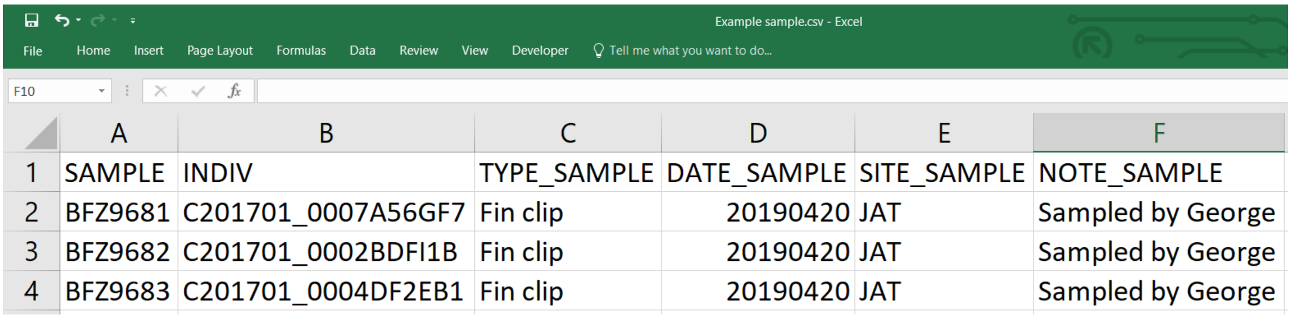
Triggering activity: ‘Tissue sampling’.

Dependencies: ‘tagging’.

Database tables populated: ‘SAMPLE’.

Column names	Description of data	Definition
SAMPLE	CHARACTER(7) UNIQUE NOT NULL	Sample identifier (label in LabCollector)
INDIV	VARCHAR(22) NOT NULL	Up to 22 characters. The first 7 characters are the spawn round identifier, the eighth character is an underscore and the final characters are the PIT tag identifier.
TYPE_SAMPLE	VARCHAR(255) NOT NULL	Sample type (e.g. Fin clip tissue; sample_type in LabCollector)
SAMPLE_BOX_1	VARCHAR(255) NOT NULL	Sample box identifier/description/location (box_id in LabCollector)
SAMPLE_BOX_2	VARCHAR(255)	Sample box identifier/description/location (secondary_box_id in LabCollector)
DATE_SAMPLE	CHARACTER(8) NOT NULL	Date (YYYYMMDD) of sampling

Table 29. Description of fields in the ‘sample’ load file.



	A	B	C	D	E	F
1	SAMPLE	INDIV	TYPE_SAMPLE	DATE_SAMPLE	SITE_SAMPLE	NOTE_SAMPLE
2	BFZ9681	C201701_0007A56GF7	Fin clip	20190420	JAT	Sampled by George
3	BFZ9682	C201701_0002BDFI1B	Fin clip	20190420	JAT	Sampled by George
4	BFZ9683	C201701_0004DF2EB1	Fin clip	20190420	JAT	Sampled by George

Figure 29. Example of a ‘sample’ load file.

1.27.site

Description: Defines sites where breeding or other activities occur (Table 30; Figure 30).

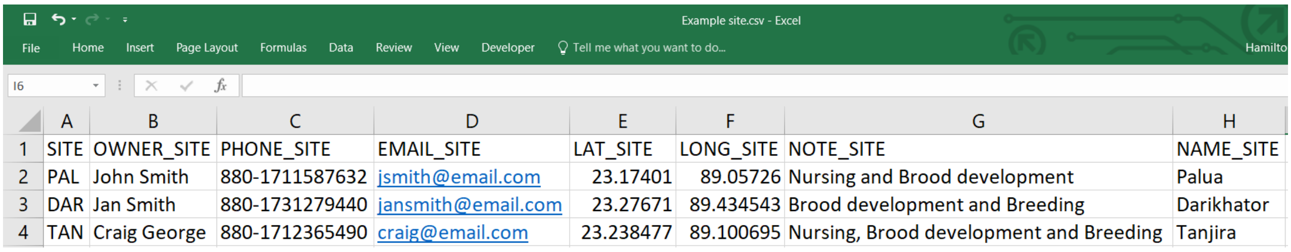
Triggering activity: ‘Nursing’, ‘Progeny testing’ and ‘Spawning’.

Dependencies: None.

Database tables populated: ‘SITE’.

Column names	Description of data	Definition
SITE	CHARACTER(3) UNIQUE NOT NULL	Three character site identifier
OWNER_SITE	VARCHAR(255) NOT NULL	Name of site owner
PHONE_SITE	VARCHAR(15) NOT NULL	Telephone contact of site owner
EMAIL_SITE	VARCHAR(255) NOT NULL	Email contact of site owner
LAT_SITE	REAL NOT NULL	Latitude in decimal degrees of main building or pond on site
LONG_SITE	REAL NOT NULL	Longitude in decimal degrees of main building or pond on site
NOTE_SITE	VARCHAR(255)	Any notable features of the site
NAME_SITE	VARCHAR(255) NOT NULL	Name of site (e.g. closest town/village identified on Google Maps)

Table 30. Description of fields in the ‘site’ load file.



	A	B	C	D	E	F	G	H
1	SITE	OWNER_SITE	PHONE_SITE	EMAIL_SITE	LAT_SITE	LONG_SITE	NOTE_SITE	NAME_SITE
2	PAL	John Smith	880-1711587632	jsmith@email.com	23.17401	89.05726	Nursing and Brood development	Palua
3	DAR	Jan Smith	880-1731279440	jansmith@email.com	23.27671	89.434543	Brood development and Breeding	Darikhator
4	TAN	Craig George	880-1712365490	craig@email.com	23.238477	89.100695	Nursing, Brood development and Breeding	Tanjira

Figure 30. Example of a ‘site’ load file.

1.28.spawn_batch

Description: Groups spawn runs within spawn rounds into spawn batches for genetic analyses (Table 31; Figure 31).

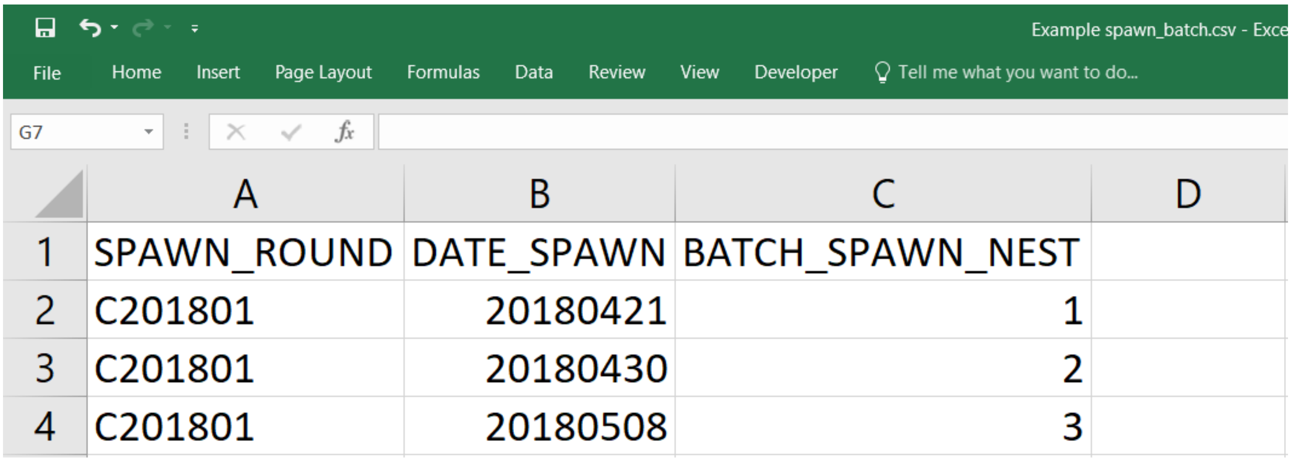
Triggering activity: ‘Spawning’.

Dependencies: ‘spawning’.

Database tables populated: ‘SPAWN_BATCH’.

Column names	Description of data	Definition
SPAWN_ROUND	CHARACTER(7) NOT NULL	7 characters. First character represents the breeding program (e.g. C for carp) next four characters are the year and the final 2 characters is the spawn run within year and breeding program
DATE_SPAWN	CHARACTER(8) NOT NULL	Spawning date (YYYYMMDD)
BATCH_SPAWN_NEST	CHARACTER(1) NOT NULL	Single character to assign spawn dates to spawn batches for analysis

Table 31. Description of fields in the ‘spawn_batch’ load file.


Example spawn_batch.csv - Exce
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Tell me what you want to do...
G7

	A	B	C	D
1	SPAWN_ROUND	DATE_SPAWN	BATCH_SPAWN_NEST	
2	C201801	20180421	1	
3	C201801	20180430	2	
4	C201801	20180508	3	

Figure 31. Example of a ‘spawn_batch’ load file.

1.29.spawn_round

Description: Identifies spawn rounds. A spawn round may include multiple species and multiple spawning batches within a spawning season. All routine family production batches of all species in a season would generally be included in a single spawn round (Table 32; Figure 32).

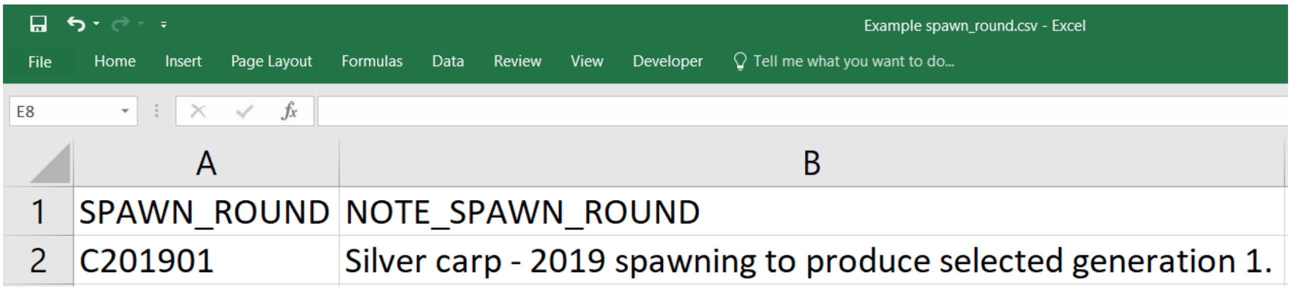
Triggering activity: 'Founders introduced', 'Nursing' and 'Spawing'.

Dependencies: None.

Database tables populated: 'SPAWN_ROUND'.

Column names	Description of data	Definition
SPAWN_ROUND	CHARACTER(7) NOT NULL	7 characters. First character represents the breeding program (e.g. C for carp) next four characters are the year and the final 2 characters is the spawn run within year and breeding program
NOTE_SPAWN_ROUND	VARCHAR(255)	Short note (e.g. place of spawning)

Table 32. Description of fields in the 'spawn_round' load file.


Example spawn_round.csv - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Tell me what you want to do...
E8
fx

	A	B
1	SPAWN_ROUND	NOTE_SPAWN_ROUND
2	C201901	Silver carp - 2019 spawning to produce selected generation 1.

Figure 32. Example of a 'spawn_round' load file.

1.30.spawning

Description: Identifies families, their parents and data generated at spawning (Table 33; Figure 33).

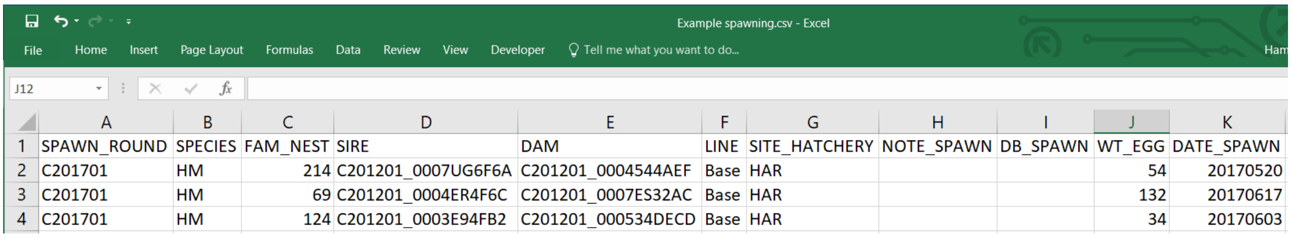
Triggering activity: ‘Spawning’.

Dependencies: ‘founder_desc’ or ‘tagging’, and ‘site’.

Database tables populated: ‘SPAWN_DATE’, ‘FAMILY’ and ‘SPAWN’.

Column names	Description of data	Definition
SPAWN_ROUND	CHARACTER(7) NOT NULL	7 characters. First character represents the breeding program (e.g. C for carp) next four characters are the year and the final 2 characters is the spawn run within year and breeding program.
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name).
FAM_NEST	VARCHAR(4) NOT NULL	Family identifier within SPAWN_ROUND and SPECIES (Integer < 10000)
SIRE	VARCHAR(22) NOT NULL	Up to 22 characters. The first 7 characters are the spawn round identifier, the eighth character is an underscore and the final characters are the PIT tag identifier.
DAM	VARCHAR(22) NOT NULL	Up to 22 characters. The first 7 characters are the spawn round identifier, the eighth character is an underscore and the final characters are the PIT tag identifier.
LINE	VARCHAR(12) NOT NULL	One of 'Base_parent; 'Base'; 'Selection'; 'Control' or 'Negative'.
SITE_HATCHERY	CHARACTER(3) NOT NULL	Three character site identifier
NOTE_SPAWN	VARCHAR(255)	Short notes remarks made at the time of spawning.
DB_SPAWN	VARCHAR(255)	Identifier in WorldFish data capture system Spawning table.
WT_EGG	REAL	Weight of eggs (grams) from the dam.
DATE_SPAWN	CHARACTER(8) NOT NULL	Spawning date (YYYYMMDD).

Table 33. Description of fields in the ‘spawning’ load file.



	A	B	C	D	E	F	G	H	I	J	K
1	SPAWN_ROUND	SPECIES	FAM_NEST	SIRE	DAM	LINE	SITE_HATCHERY	NOTE_SPAWN	DB_SPAWN	WT_EGG	DATE_SPAWN
2	C201701	HM	214	C201201_0007UG6F6A	C201201_0004544AEF	Base	HAR			54	20170520
3	C201701	HM	69	C201201_0004ER4F6C	C201201_0007ES32AC	Base	HAR			132	20170617
4	C201701	HM	124	C201201_0003E94FB2	C201201_000534DECD	Base	HAR			34	20170603

Figure 33. Example of a ‘spawning’ load file.

1.31.species

Description: Enters species details (Table 34; Figure 34).

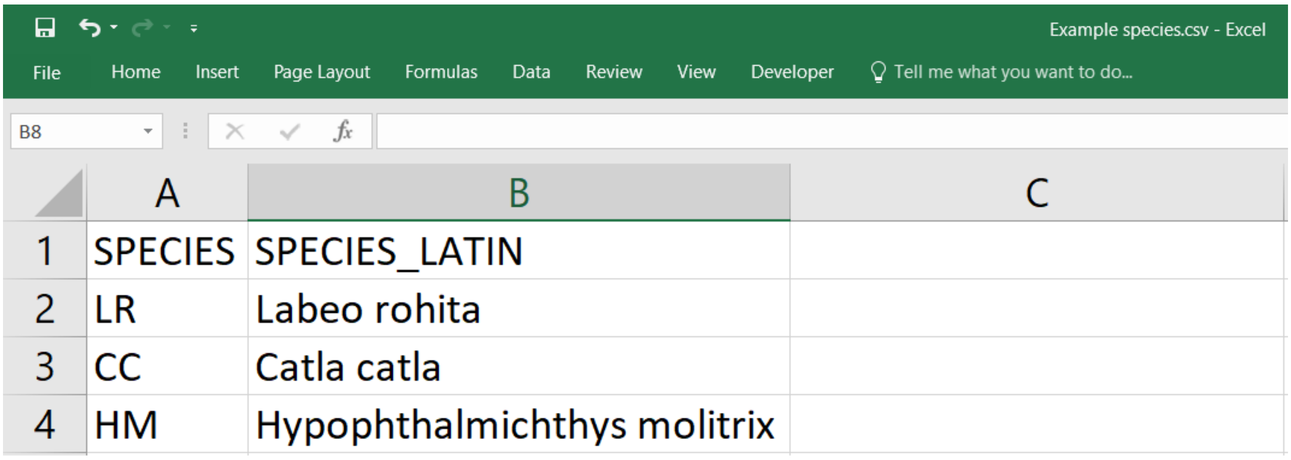
Triggering activity: ‘Founders introduced’, ‘Genetic analysis’ and ‘Progeny testing’.

Dependencies: None.

Database tables populated: ‘SPECIES’.

Column names	Description of data	Definition
SPECIES	CHARACTER(2) UNIQUE NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
SPECIES_LATIN	VARCHAR(255) UNIQUE NOT NULL	Scientific name of species

Table 34. Description of fields in the ‘species’ load file.



	A	B	C
1	SPECIES	SPECIES_LATIN	
2	LR	Labeo rohita	
3	CC	Catla catla	
4	HM	Hypophthalmichthys molitrix	

Figure 34. Example of a ‘species’ load file.

1.32.tagging

Description: Assigns individual identifiers (and families) to fish. If the family is unknown at the time of 'tagging' then the FAM_NEST column should be left blank. The family can then be assigned at a later time using 'family_assign' load (Table 35; Figure 35).

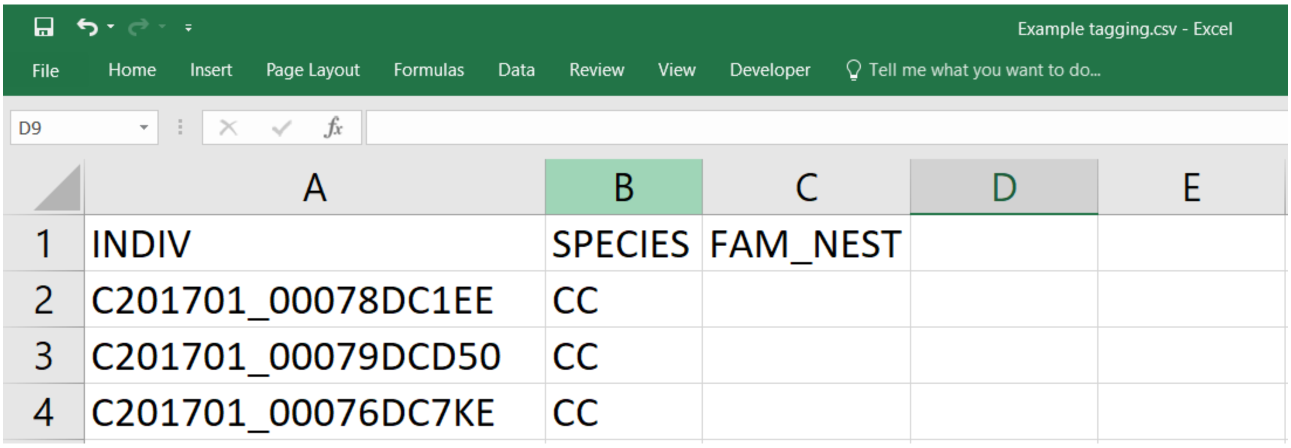
Triggering activity: 'Tagging / family assignment'.

Dependencies: 'spawning'.

Database tables populated: 'INDIV_PIT', 'INDIV_SPECIES' and 'INDIV_FAM'.

Column names	Description of data	Definition
INDIV	VARCHAR(22) NOT NULL	Up to 22 characters. The first 7 characters are the spawn round identifier, the eighth character is an underscore and the final characters are the PIT tag identifier.
SPECIES	CHARACTER(2) NOT NULL	Two uppercase letter identifier for species (e.g. acronym of scientific name)
FAM_NEST	INTEGER(4)	Family identifier within SPAWN_ROUND and SPECIES (Integer < 10000)

Table 35. Description of fields in 'tagging' load files.



	A	B	C	D	E
1	INDIV	SPECIES	FAM_NEST		
2	C201701_00078DC1EE	CC			
3	C201701_00079DCD50	CC			
4	C201701_00076DC7KE	CC			

Figure 35. Example of a 'tagging' load file.

1.33.tbl_details

Description: Describes the tables in the GCIP Database (Table 36; Figure 36).

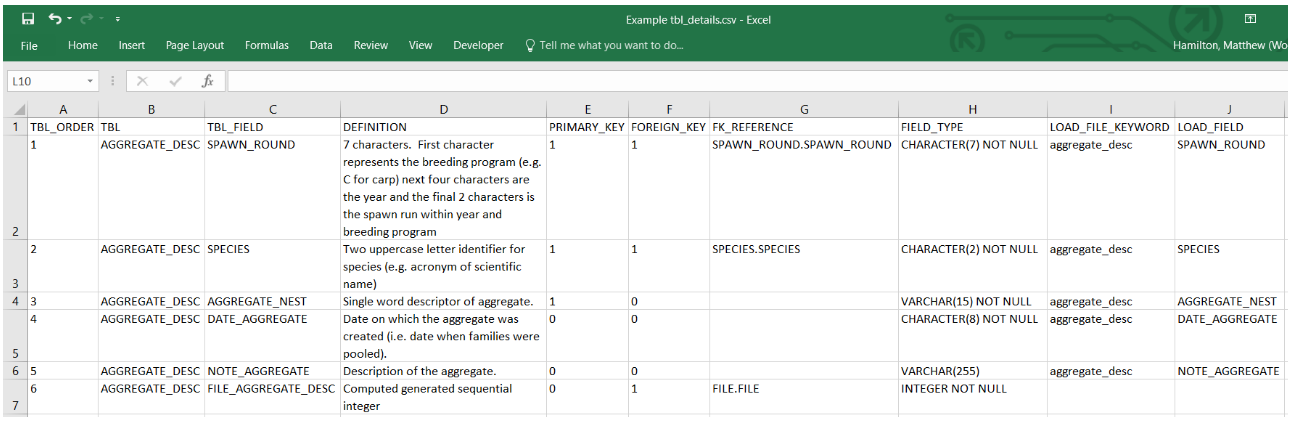
Triggering activity: ‘Database described’.

Dependencies: None.

Database tables populated: ‘TBL_DETAILS’.

Column names	Description of data	Definition
TBL_ORDER	INTEGER UNIQUE NOT NULL	Unique identifier (integer)
TBL	VARCHAR(255) NOT NULL	Table name
TBL_FIELD	VARCHAR(255) NOT NULL	Table field
DEFINITION	VARCHAR(255) NOT NULL	Definition of table field
PRIMARY_KEY	VARCHAR(5) NOT NULL	TRUE or FALSE
FOREIGN_KEY	FOREIGN_KEY	TRUE or FALSE
FK_REFERENCE	VARCHAR(255)	Reference to foreign key (Table.Table_field format)
FIELD_TYPE	VARCHAR(255) NOT NULL	Field constraints
LOAD_FILE_KEYWORD	VARCHAR(255)	File name of load file
LOAD_FIELD	VARCHAR(255)	Field name in load file

Table 36. Description of fields in the ‘tbl_details’ load file.



	A	B	C	D	E	F	G	H	I	J
1	TBL_ORDER	TBL	TBL_FIELD	DEFINITION	PRIMARY_KEY	FOREIGN_KEY	FK_REFERENCE	FIELD_TYPE	LOAD_FILE_KEYWORD	LOAD_FIELD
2	1	AGGREGATE_DESC	SPAWN_ROUND	7 characters. First character represents the breeding program (e.g. C for carp) next four characters are the year and the final 2 characters is the spawn run within year and breeding program	1	1	SPAWN_ROUND.SPAWN_ROUND	CHARACTER(7) NOT NULL	aggregate_desc	SPAWN_ROUND
3	2	AGGREGATE_DESC	SPECIES	Two uppercase letter identifier for species (e.g. acronym of scientific name)	1	1	SPECIES.SPECIES	CHARACTER(2) NOT NULL	aggregate_desc	SPECIES
4	3	AGGREGATE_DESC	AGGREGATE_NEST	Single word descriptor of aggregate.	1	0		VARCHAR(15) NOT NULL	aggregate_desc	AGGREGATE_NEST
5	4	AGGREGATE_DESC	DATE_AGGREGATE	Date on which the aggregate was created (i.e. date when families were pooled).	0	0		CHARACTER(8) NOT NULL	aggregate_desc	DATE_AGGREGATE
6	5	AGGREGATE_DESC	NOTE_AGGREGATE	Description of the aggregate.	0	0		VARCHAR(255)	aggregate_desc	NOTE_AGGREGATE
7	6	AGGREGATE_DESC	FILE_AGGREGATE_DESC	Computed generated sequential integer	0	1	FILE.FILE	INTEGER NOT NULL		

Figure 36. Example of a ‘tbl_details’ load file.

1.34.trait

Description: Describes the traits measured on individual tagged fish (Table 37; Figure 37).

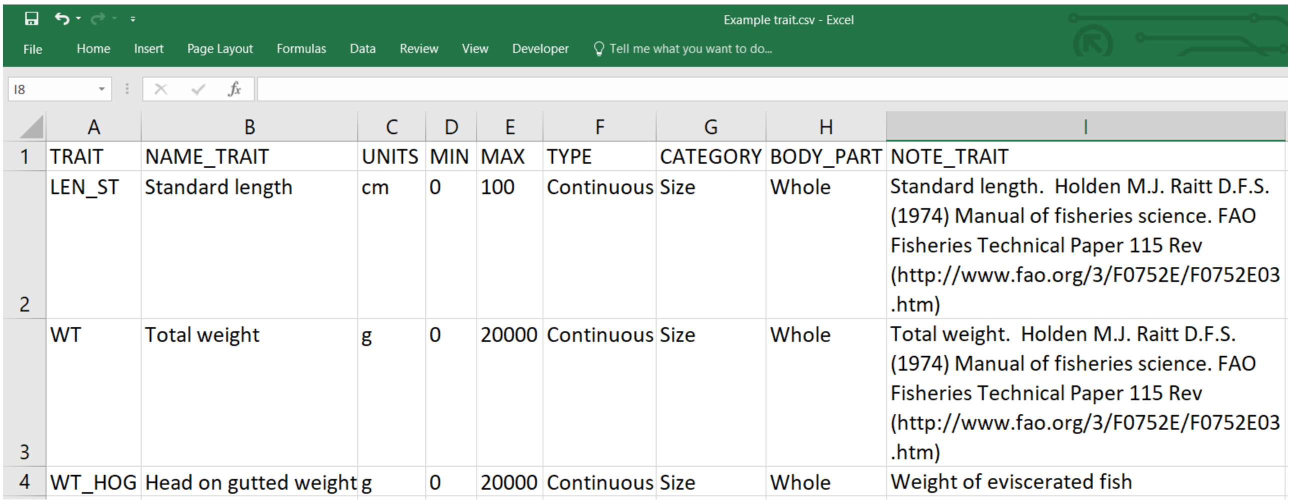
Triggering activity: ‘Measurement’ and ‘Genetic analysis’.

Dependencies: None.

Database tables populated: ‘TRAIT’.

Column names	Description of data	Definition
TRAIT	VARCHAR(10) UNIQUE NOT NULL	Short trait name/identifier
NAME_TRAIT	VARCHAR(40) NOT NULL	Long trait name/identifier
UNITS	VARCHAR(40) NOT NULL	Units in which trait is measured
MIN	REAL	Minimum possible value of trait (may be blank)
MAX	REAL	Maximum possible value of trait (may be blank)
TYPE	VARCHAR(20) NOT NULL	One of 'Continuous'; 'Categorical'; 'Ordinal' or 'Binary'
CATEGORY	VARCHAR(20) NOT NULL	One of 'Bones'; 'Colour'; 'Feed'; 'Fitness' 'Flesh'; 'Health'; 'Nutrient'; 'Sex'; 'Size' or 'Other'
BODY_PART	VARCHAR(20) NOT NULL	One of 'Anus'; 'Body'; 'Eyes'; 'Fins'; 'Flesh'; 'Gall'; 'Gills'; 'Gonads'; 'Head'; 'Kidney'; 'Liver'; 'Opercula'; 'Spleen'; 'Viscera'; 'Whole' or 'Other'
NOTE_TRAIT	VARCHAR(255) NOT NULL	Notes/comments on trait

Table 37. Description of fields in the ‘trait’ load file.



	A	B	C	D	E	F	G	H	I
1	TRAIT	NAME_TRAIT	UNITS	MIN	MAX	TYPE	CATEGORY	BODY_PART	NOTE_TRAIT
2	LEN_ST	Standard length	cm	0	100	Continuous	Size	Whole	Standard length. Holden M.J. Raitt D.F.S. (1974) Manual of fisheries science. FAO Fisheries Technical Paper 115 Rev (http://www.fao.org/3/F0752E/F0752E03.htm)
3	WT	Total weight	g	0	20000	Continuous	Size	Whole	Total weight. Holden M.J. Raitt D.F.S. (1974) Manual of fisheries science. FAO Fisheries Technical Paper 115 Rev (http://www.fao.org/3/F0752E/F0752E03.htm)
4	WT_HOG	Head on gutted weight	g	0	20000	Continuous	Size	Whole	Weight of eviscerated fish

Figure 37. Example of a ‘trait’ load file.

1.35.treatment

Description: Describes culture, fertiliser, feed and treatment applied to ponds (e.g. progeny test ponds; Table 37; Figure 38).

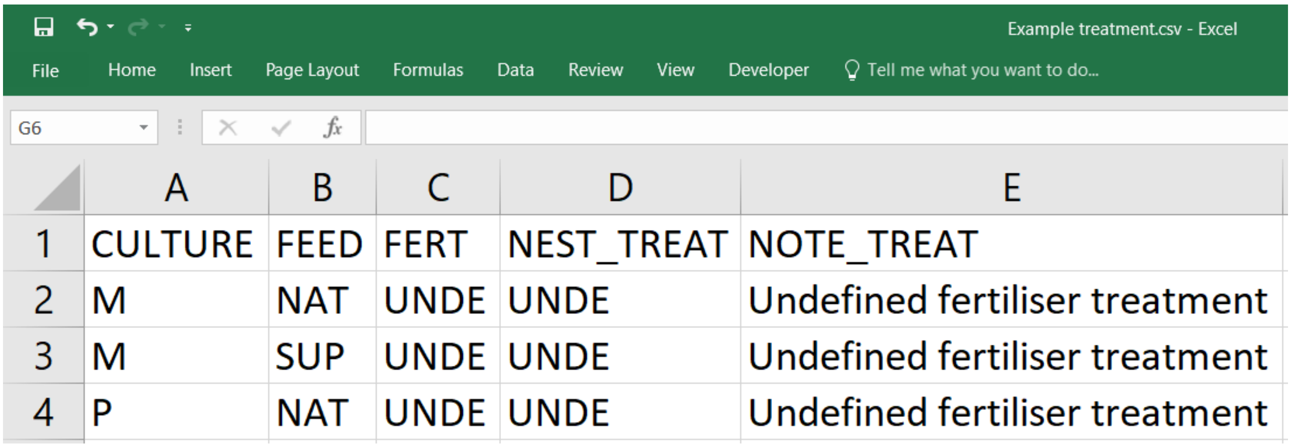
Triggering activity: ‘Genetic analysis’ and ‘Progeny testing’.

Dependencies: ‘feed’ and ‘fert’.

Database tables populated: ‘TREATMENT’.

Column names	Description of data	Definition
CULTURE	CHARACTER(1) NOT NULL	‘M’ monoculture or ‘P’ for polyculture
FERT	CHARACTER(4) NOT NULL	Fertiliser identifier
FEED	CHARACTER(3) NOT NULL	Feed identifier
NEST_TREAT	CHARACTER(4) NOT NULL	Treatment identifier
NOTE_TREAT	VARCHAR(255)	Note on treatment (e.g. medicines, chemical applications)

Table 38. Description of fields in the ‘treatment’ load file.



	A	B	C	D	E
1	CULTURE	FEED	FERT	NEST_TREAT	NOTE_TREAT
2	M	NAT	UNDE	UNDE	Undefined fertiliser treatment
3	M	SUP	UNDE	UNDE	Undefined fertiliser treatment
4	P	NAT	UNDE	UNDE	Undefined fertiliser treatment

Figure 38. Example of a ‘treatment’ load file.

4. References

Chang W, Cheng J, Allaire J, Sievert C, Schloerke B, Xie Y, Allen J, McPherson J, Dipert A and Borges B. 2021. shiny: Web Application Framework for R. R package version 1.6.0.

Hamilton M, Rajts F, Collis W, Shanta SM, Alam MB and Kabir MS. 2020. WorldFish Carp Genetic Improvement Program Electronic Pond Book Guide. Penang, Malaysia: WorldFish. Handbook. https://doi.org/10.7910/DVN/DBN1UD

Hamilton MG, Mekkawy W and Benzie JAH. 2019a. Sibship assignment to the founders of a Bangladeshi *Catla catla* breeding population. Genetics Selection Evolution. 51, 17. https://doi.org/10.1186/s12711-019-0454-x

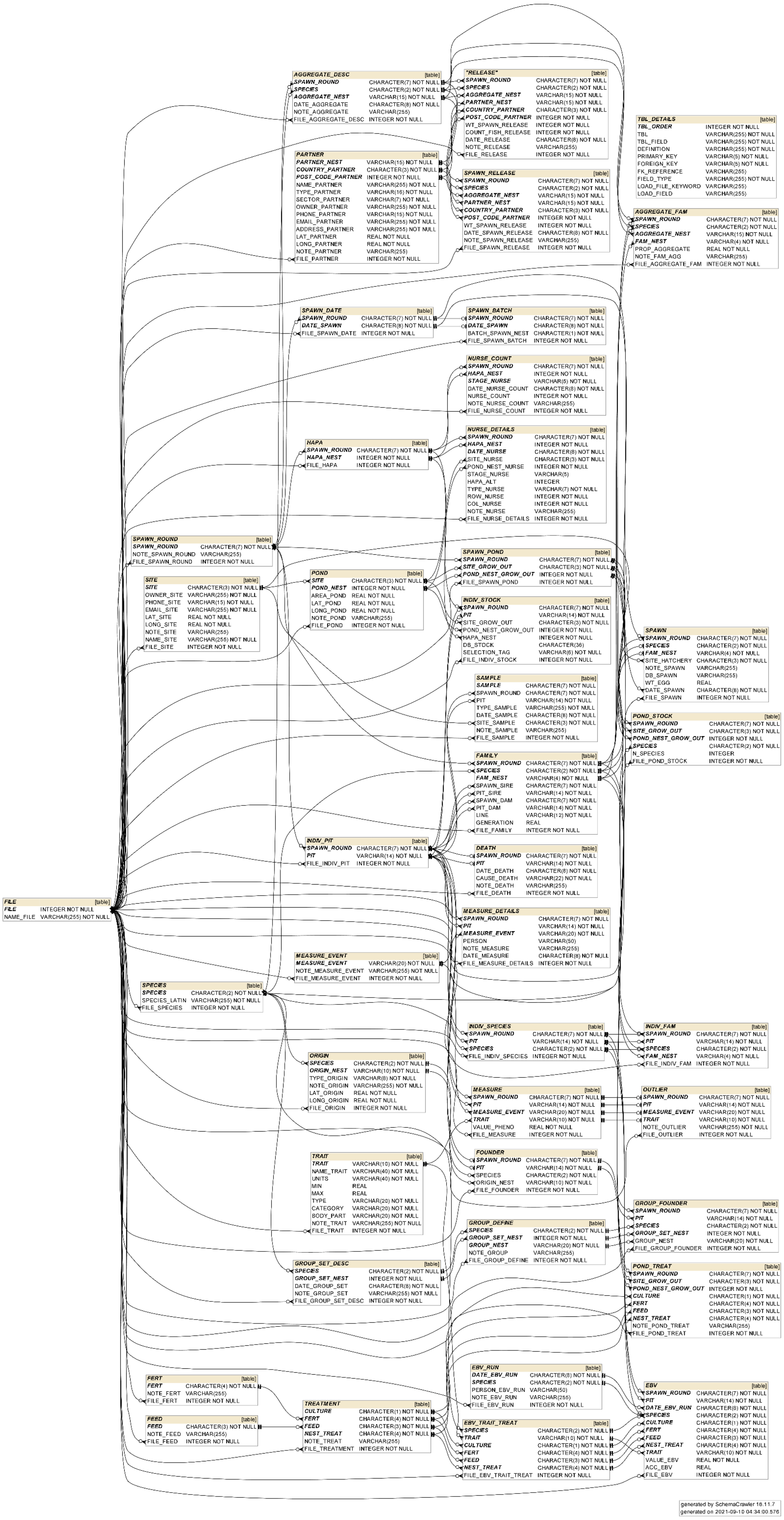
Hamilton MG, Mekkawy W, Kilian A and Benzie JAH. 2019b. Single Nucleotide Polymorphisms (SNPs) reveal sibship among founders of a Bangladeshi rohu (*Labeo rohita*) breeding population. Frontiers in Genetics. 10. https://doi.org/10.3389/fgene.2019.00597

Hamilton MG, Mekkawy W, Barman BK, Alam MB, Karim M and Benzie JAH. 2021. Genetic relationships among founders of a silver carp (*Hypophthalmichthys molitrix*) genetic improvement program in Bangladesh. Aquaculture, 736715. https://doi.org/10.1016/j.aquaculture.2021.736715

Müller K, Wickham H, James DA and Falcon S. 2021. RSQLite: 'SQLite' Interface for R. R package version 2.2.3.

R Core Team, 2020. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria

Annex 1. WorldFish CGIP Database schema



AGGREGATE_DESC	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
SPECIES	CHARACTER(2) NOT NULL
AGGREGATE_NEST	VARCHAR(15) NOT NULL
DATE_AGGREGATE	CHARACTER(8) NOT NULL
NOTE_AGGREGATE	VARCHAR(255)
FILE_AGGREGATE_DESC	INTEGER NOT NULL

"RELEASE"	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
SPECIES	CHARACTER(2) NOT NULL
AGGREGATE_NEST	VARCHAR(15) NOT NULL
PARTNER_NEST	VARCHAR(15) NOT NULL
COUNTRY_PARTNER	CHARACTER(3) NOT NULL
POST_CODE_PARTNER	INTEGER NOT NULL
WT_SPAWN_RELEASE	INTEGER NOT NULL
COUNT_FISH_RELEASE	INTEGER NOT NULL
DATE_RELEASE	CHARACTER(8) NOT NULL
NOTE_RELEASE	VARCHAR(255)
FILE_RELEASE	INTEGER NOT NULL

TBL_DETAILS	[table]
TBL_ORDER	INTEGER NOT NULL
TBL	VARCHAR(255) NOT NULL
TBL_FIELD	VARCHAR(255) NOT NULL
DEFINITION	VARCHAR(255) NOT NULL
PRIMARY_KEY	VARCHAR(5) NOT NULL
FOREIGN_KEY	VARCHAR(5) NOT NULL
FK_REFERENCE	VARCHAR(255)
FIELD_TYPE	VARCHAR(255) NOT NULL
LOAD_FILE_KEYWORD	VARCHAR(255)
LOAD_FIELD	VARCHAR(255)

PARTNER	[table]
PARTNER_NEST	VARCHAR(15) NOT NULL
COUNTRY_PARTNER	CHARACTER(3) NOT NULL
POST_CODE_PARTNER	INTEGER NOT NULL
NAME_PARTNER	VARCHAR(255) NOT NULL
TYPE_PARTNER	VARCHAR(16) NOT NULL
SECTOR_PARTNER	VARCHAR(7) NOT NULL
OWNER_PARTNER	VARCHAR(255) NOT NULL
PHONE_PARTNER	VARCHAR(15) NOT NULL
EMAIL_PARTNER	VARCHAR(255) NOT NULL
ADDRESS_PARTNER	VARCHAR(255) NOT NULL
LAT_PARTNER	REAL NOT NULL
LONG_PARTNER	REAL NOT NULL
NOTE_PARTNER	VARCHAR(255)
FILE_PARTNER	INTEGER NOT NULL

SPAWN_RELEASE	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
SPECIES	CHARACTER(2) NOT NULL
AGGREGATE_NEST	VARCHAR(15) NOT NULL
PARTNER_NEST	VARCHAR(15) NOT NULL
COUNTRY_PARTNER	CHARACTER(3) NOT NULL
POST_CODE_PARTNER	INTEGER NOT NULL
WT_SPAWN_RELEASE	INTEGER NOT NULL
DATE_SPAWN_RELEASE	CHARACTER(8) NOT NULL
NOTE_SPAWN_RELEASE	VARCHAR(255)
FILE_SPAWN_RELEASE	INTEGER NOT NULL

AGGREGATE_FAM	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
SPECIES	CHARACTER(2) NOT NULL
AGGREGATE_NEST	VARCHAR(15) NOT NULL
FAM_NEST	VARCHAR(4) NOT NULL
PROP_AGGREGATE	REAL NOT NULL
NOTE_FAM_AGG	VARCHAR(255)
FILE_AGGREGATE_FAM	INTEGER NOT NULL

SPAWN_DATE	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
DATE_SPAWN	CHARACTER(8) NOT NULL
FILE_SPAWN_DATE	INTEGER NOT NULL

SPAWN_BATCH	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
DATE_SPAWN	CHARACTER(8) NOT NULL
BATCH_SPAWN_NEST	CHARACTER(1) NOT NULL
FILE_SPAWN_BATCH	INTEGER NOT NULL

NURSE_COUNT	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
HAPA_NEST	INTEGER NOT NULL
STAGE_NURSE	VARCHAR(5) NOT NULL
DATE_NURSE_COUNT	CHARACTER(8) NOT NULL
NURSE_COUNT	INTEGER NOT NULL
NOTE_NURSE_COUNT	VARCHAR(255)
FILE_NURSE_COUNT	INTEGER NOT NULL

NURSE_DETAILS	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
HAPA_NEST	INTEGER NOT NULL
DATE_NURSE	CHARACTER(8) NOT NULL
SITE_NURSE	CHARACTER(3) NOT NULL
POND_NEST_NURSE	INTEGER NOT NULL
STAGE_NURSE	VARCHAR(5)
HAPA_ALT	INTEGER
TYPE_NURSE	VARCHAR(7) NOT NULL
ROW_NURSE	INTEGER NOT NULL
COL_NURSE	INTEGER NOT NULL
NOTE_NURSE	VARCHAR(255)
FILE_NURSE_DETAILS	INTEGER NOT NULL

HAPA	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
HAPA_NEST	INTEGER NOT NULL
FILE_HAPA	INTEGER NOT NULL

SPAWN_ROUND	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
NOTE_SPAWN_ROUND	VARCHAR(255)
FILE_SPAWN_ROUND	INTEGER NOT NULL

SPAWN_POND	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
SITE_GROW_OUT	CHARACTER(3) NOT NULL
POND_NEST_GROW_OUT	INTEGER NOT NULL
FILE_SPAWN_POND	INTEGER NOT NULL

SITE	[table]
SITE	CHARACTER(3) NOT NULL
OWNER_SITE	VARCHAR(255) NOT NULL
PHONE_SITE	VARCHAR(15) NOT NULL
EMAIL_SITE	VARCHAR(255) NOT NULL
LAT_SITE	REAL NOT NULL
LONG_SITE	REAL NOT NULL
NOTE_SITE	VARCHAR(255)
NAME_SITE	VARCHAR(255) NOT NULL
FILE_SITE	INTEGER NOT NULL

POND	[table]
SITE	CHARACTER(3) NOT NULL
POND_NEST	INTEGER NOT NULL
AREA_POND	REAL NOT NULL
LAT_POND	REAL NOT NULL
LONG_POND	REAL NOT NULL
NOTE_POND	VARCHAR(255)
FILE_POND	INTEGER NOT NULL

INDIV_STOCK	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
PIT	VARCHAR(14) NOT NULL
SITE_GROW_OUT	CHARACTER(3) NOT NULL
POND_NEST_GROW_OUT	INTEGER NOT NULL
HAPA_NEST	INTEGER NOT NULL
DB_STOCK	CHARACTER(36)
SELECTION_TAG	VARCHAR(6) NOT NULL
FILE_INDIV_STOCK	INTEGER NOT NULL

SPAWN	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
SPECIES	CHARACTER(2) NOT NULL
FAM_NEST	VARCHAR(4) NOT NULL
SITE_HATCHERY	CHARACTER(3) NOT NULL
NOTE_SPAWN	VARCHAR(255)
DB_SPAWN	VARCHAR(255)
WT_EGG	REAL
DATE_SPAWN	CHARACTER(8) NOT NULL
FILE_SPAWN	INTEGER NOT NULL

SAMPLE	[table]
SAMPLE	CHARACTER(7) NOT NULL
SPAWN_ROUND	CHARACTER(7) NOT NULL
PIT	VARCHAR(14) NOT NULL
TYPE_SAMPLE	VARCHAR(255) NOT NULL
DATE_SAMPLE	CHARACTER(8) NOT NULL
SITE_SAMPLE	CHARACTER(3) NOT NULL
NOTE_SAMPLE	VARCHAR(255)
FILE_SAMPLE	INTEGER NOT NULL

POND_STOCK	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
SITE_GROW_OUT	CHARACTER(3) NOT NULL
POND_NEST_GROW_OUT	INTEGER NOT NULL
SPECIES	CHARACTER(2) NOT NULL
N_SPECIES	INTEGER
FILE_POND_STOCK	INTEGER NOT NULL

FAMILY	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
SPECIES	CHARACTER(2) NOT NULL
FAM_NEST	VARCHAR(4) NOT NULL
SPAWN_SIRE	CHARACTER(7) NOT NULL
PIT_SIRE	VARCHAR(14) NOT NULL
SPAWN_DAM	CHARACTER(7) NOT NULL
PIT_DAM	VARCHAR(14) NOT NULL
LINE	VARCHAR(12) NOT NULL
GENERATION	REAL
FILE_FAMILY	INTEGER NOT NULL

INDIV_PIT	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
PIT	VARCHAR(14) NOT NULL
FILE_INDIV_PIT	INTEGER NOT NULL

DEATH	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
PIT	VARCHAR(14) NOT NULL
DATE_DEATH	CHARACTER(8) NOT NULL
CAUSE_DEATH	VARCHAR(22) NOT NULL
NOTE_DEATH	VARCHAR(255)
FILE_DEATH	INTEGER NOT NULL

FILE	[table]
FILE	INTEGER NOT NULL
NAME_FILE	VARCHAR(255) NOT NULL

MEASURE_DETAILS	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
PIT	VARCHAR(14) NOT NULL
MEASURE_EVENT	VARCHAR(20) NOT NULL
PERSON	VARCHAR(50)
NOTE_MEASURE	VARCHAR(255)
DATE_MEASURE	CHARACTER(8) NOT NULL
FILE_MEASURE_DETAILS	INTEGER NOT NULL

MEASURE_EVENT	[table]
MEASURE_EVENT	VARCHAR(20) NOT NULL
NOTE_MEASURE_EVENT	VARCHAR(255) NOT NULL
FILE_MEASURE_EVENT	INTEGER NOT NULL

SPECIES	[table]
SPECIES	CHARACTER(2) NOT NULL
SPECIES_LATIN	VARCHAR(255) NOT NULL
FILE_SPECIES	INTEGER NOT NULL

INDIV_SPECIES	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
PIT	VARCHAR(14) NOT NULL
SPECIES	CHARACTER(2) NOT NULL
FILE_INDIV_SPECIES	INTEGER NOT NULL

INDIV_FAM	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
PIT	VARCHAR(14) NOT NULL
SPECIES	CHARACTER(2) NOT NULL
FAM_NEST	VARCHAR(4) NOT NULL
FILE_INDIV_FAM	INTEGER NOT NULL

ORIGIN	[table]
SPECIES	CHARACTER(2) NOT NULL
ORIGIN_NEST	VARCHAR(10) NOT NULL
TYPE_ORIGIN	VARCHAR(8) NOT NULL
NOTE_ORIGIN	VARCHAR(255) NOT NULL
LAT_ORIGIN	REAL NOT NULL
LONG_ORIGIN	REAL NOT NULL
FILE_ORIGIN	INTEGER NOT NULL

MEASURE	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
PIT	VARCHAR(14) NOT NULL
MEASURE_EVENT	VARCHAR(20) NOT NULL
TRAIT	VARCHAR(10) NOT NULL
VALUE_PHENO	REAL NOT NULL
FILE_MEASURE	INTEGER NOT NULL

OUTLIER	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
PIT	VARCHAR(14) NOT NULL
MEASURE_EVENT	VARCHAR(20) NOT NULL
TRAIT	VARCHAR(10) NOT NULL
NOTE_OUTLIER	VARCHAR(255) NOT NULL
FILE_OUTLIER	INTEGER NOT NULL

TRAIT	[table]
TRAIT	VARCHAR(10) NOT NULL
NAME_TRAIT	VARCHAR(40) NOT NULL
UNITS	VARCHAR(40) NOT NULL
MIN	REAL
MAX	REAL
TYPE	VARCHAR(20) NOT NULL
CATEGORY	VARCHAR(20) NOT NULL
BODY_PART	VARCHAR(20) NOT NULL
NOTE_TRAIT	VARCHAR(255) NOT NULL
FILE_TRAIT	INTEGER NOT NULL

FOUNDER	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
PIT	VARCHAR(14) NOT NULL
SPECIES	CHARACTER(2) NOT NULL
ORIGIN_NEST	VARCHAR(10) NOT NULL
FILE_FOUNDER	INTEGER NOT NULL

GROUP_FOUNDER	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
PIT	VARCHAR(14) NOT NULL
SPECIES	CHARACTER(2) NOT NULL
GROUP_SET_NEST	INTEGER NOT NULL
GROUP_NEST	VARCHAR(20) NOT NULL
FILE_GROUP_FOUNDER	INTEGER NOT NULL

GROUP_DEFINE	[table]
SPECIES	CHARACTER(2) NOT NULL
GROUP_SET_NEST	INTEGER NOT NULL
GROUP_NEST	VARCHAR(20) NOT NULL
NOTE_GROUP	VARCHAR(255)
FILE_GROUP_DEFINE	INTEGER NOT NULL

GROUP_SET_DESC	[table]
SPECIES	CHARACTER(2) NOT NULL
GROUP_SET_NEST	INTEGER NOT NULL
DATE_GROUP_SET	CHARACTER(8) NOT NULL
NOTE_GROUP_SET	VARCHAR(255) NOT NULL
FILE_GROUP_SET_DESC	INTEGER NOT NULL

POND_TREAT	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
SITE_GROW_OUT	CHARACTER(3) NOT NULL
POND_NEST_GROW_OUT	INTEGER NOT NULL
CULTURE	CHARACTER(1) NOT NULL
FERT	CHARACTER(4) NOT NULL
FEED	CHARACTER(3) NOT NULL
NEST_TREAT	CHARACTER(4) NOT NULL
NOTE_POND_TREAT	VARCHAR(255)
FILE_POND_TREAT	INTEGER NOT NULL

EBV_RUN	[table]
DATE_EBV_RUN	CHARACTER(8) NOT NULL
SPECIES	CHARACTER(2) NOT NULL
PERSON_EBV_RUN	VARCHAR(50)
NOTE_EBV_RUN	VARCHAR(255)
FILE_EBV_RUN	INTEGER NOT NULL

EBV	[table]
SPAWN_ROUND	CHARACTER(7) NOT NULL
PIT	VARCHAR(14) NOT NULL
DATE_EBV_RUN	CHARACTER(8) NOT NULL
SPECIES	CHARACTER(2) NOT NULL
CULTURE	CHARACTER(1) NOT NULL
FERT	CHARACTER(4) NOT NULL
FEED	CHARACTER(3) NOT NULL
NEST_TREAT	CHARACTER(4) NOT NULL
TRAIT	VARCHAR(10) NOT NULL
VALUE_EBV	REAL NOT NULL
ACC_EBV	REAL
FILE_EBV	INTEGER NOT NULL

FERT	[table]
FERT	CHARACTER(4) NOT NULL
NOTE_FERT	VARCHAR(255)
FILE_FERT	INTEGER NOT NULL

FEED	[table]
FEED	CHARACTER(3) NOT NULL
NOTE_FEED	VARCHAR(255)
FILE_FEED	INTEGER NOT NULL

TREATMENT	[table]
CULTURE	CHARACTER(1) NOT NULL
FERT	CHARACTER(4) NOT NULL
FEED	CHARACTER(3) NOT NULL
NEST_TREAT	CHARACTER(4) NOT NULL
NOTE_TREAT	VARCHAR(255)
FILE_TREATMENT	INTEGER NOT NULL

EBV_TRAIT_TREAT	[table]
SPECIES	CHARACTER(2) NOT NULL
TRAIT	VARCHAR(10) NOT NULL
CULTURE	CHARACTER(1) NOT NULL
FERT	CHARACTER(4) NOT NULL
FEED	CHARACTER(3) NOT NULL
NEST_TREAT	CHARACTER(4) NOT NULL
FILE_EBV_TRAIT_TREAT	INTEGER NOT NULL

generated by SchemaCrawler 16.11.7
generated on 2021-09-10 04:34:00.576

WorldFish

About WorldFish
WorldFish is a nonprofit research and innovation institution that creates, advances and translates scientific research on aquatic food systems into scalable solutions with transformational impact on human well-being and the environment. Our research data, evidence and insights shape better practices, policies and investment decisions for sustainable development in low- and middle-income countries.

We have a global presence across 20 countries in Asia, Africa and the Pacific with 460 staff of 30 nationalities deployed where the greatest sustainable development challenges can be addressed through holistic aquatic food systems solutions.

Our research and innovation work spans climate change, food security and nutrition, sustainable fisheries and aquaculture, the blue economy and ocean governance, One Health, genetics and AgriTech, and it integrates evidence and perspectives on gender, youth and social inclusion. Our approach empowers people for change over the long term: research excellence and engagement with national and international partners are at the heart of our efforts to set new agendas, build capacities and support better decision-making on the critical issues of our times.

WorldFish is part of One CGIAR, the world’s largest agricultural innovation network.

For more information, please visit www.worldfishcenter.org