Wheatgenetics Database Quick Start Guide

Table of Contents

Purpose:

This user guide provides an overview of the Wheatgenetics database. It is intended to get the reader familiar with naming conventions used in the Poland lab and able to curate their own data into the database for most of the common columns and tables that will be used. In addition to providing keys to the database it will also explain many of the rationales behind the data storage procedure helping create and store data not just for the immediate project but also maintain the data and its integrity well into the future. With the exception of the genomic data tables, the phenotypic data tables have been developed for wheat. If your project stores other species data please contact Mark Lucas, mlucas@ksu.edu, before storing data. This guide assumes the reader is using a graphical user interface (GUI) such as Navicat or MySQL Workbench. Finally, this guide is not a detailed account of each table, foreign key pairs, and triggers that run behind the scene keeping the data organized.

Current Version:

This version was last updated October 29, 2016.

I have data now what?

This is the most likely place that many people will begin to utilize the database. Perhaps the field season has begun, maybe some data has been collected, the question becomes how to move data from a variety of spreadsheets into the database. Where should I start? The best place would be the experiment table.

Experiment Table:

The experiment table keeps tracks of all experiments that have been performed within the Poland lab. This table gives a high level over view of experiments over many years (As of 2016 there were over 6 years of experiments and 80 different experiments). While this table is small, a user can visually inspect and decide what experiments that they would like to query data for. Completing the experiment table will "tag" all of your data so that by simply knowing the experiment_id you can find all records associated with your experiment. The columns and guides for defining them are:

Record_id: primary key for the experiment table. This is an auto-incrementing number as and is filled out automatically.

Experiment_id: foreign key and the key to all data related to experiment. **To make your experiment_id:** Experiment_id is made of three parts separated by an underscore. The parts are harvest date year as a two digit number _ location traditionally as a three letters (although often four have been used) _ experiment name that describes the experiment. Thus 13_OBR_SynOp is a trial that was harvested in 2013, at OBRegon, Mexico, and the trial was the Synthetic Opata population. **Note:** the year is when the trial is harvested so for Kansas wheat planted in the fall of 2016, the experiment_id will be 17_ASH_YourTrialHere.

Additional notes about experiment_id: Each experiment should have its own experiment_id, even though an experiment can be repeated over time and space one experiment trial per experiment_id. For experiments repeated over time and space update the year and location to identify them, thus 16_ASH_BYD and 15_ASH_BYD are both BYD trials occurring at two different years.

Location: Takes the *year _ location* information from experiment_id and combines them without the underscore. Example 13OBR is 2013 OBRegon trial.

Environment: For trials that are repeated at the same location and year but under different treatment structure regimens. For example, trials at CIMMYT are often planted at the same year (15) at the same location (OBR) but then are planted under drought, heat, irrigated, limited irrigation.

Planting_date: The date the experiment was planted.

Harvest_date: The day harvest occurred.

C1_1_x, C1_1_y, C1_1_z – C2_2_x, C2_2_y, C2_2_z: These are UTM coordinates of a square that encloses the entire experiment area. Elements ending in x are east-west (Easting), y is north-south (Northing), and z is elevation with units all being in meters. See the plot_map table (Figure 1) for diagram of example coordinates, additionally the experiment does not have to align to a N-S or E-W axis.

Lat_zone: UTM latitude zone where the experiment lies.

Long_zone: UTM longitude zone grid in which the experiment lies.

Notes: Any additional notes about the experiment. Useful information would be type of experimental design such as randomized complete block design or alpha lattice, purpose of the experiment for example: "BYD trial is designed to test genotype resistance to barley yellow dwarf. The trial consist of 3 treated and untreated replicates that prevented/allowed aphids and the potential for BYD virus to assess the impact on grain yield."

Queries based on the experiment table:

1. Find all trials that contain BYD.

```
Select experiment.*
from experiment
where experiment.experiment_id like '%BYD%'
```

The results show there are six experiments that are related to BYD. This type of query could also be used to search the notes or other columns for a desired trait.
14_ASH_BYD
13_ASH_BYD
15_ASH_BYD
15_ASH_BYDArtXEverest
15_ASH_BYDInsecticideTrial
16_RF_BYD

2. Find plots that belong to a certain experiment. This allows one foreign key to identify all plots associated with an experiment.

```
Select plot.*
From plot
Where plot.experiment_id like '%FAM8%'
```

Returns 2016 records spanning 3 years and 2 environments. Provides a good way to query plot table without having to know exactly how the plot was identified.

```
Select plot.*
From plot
Where plot.experiment_id like '15%FAM8%'
```

Returns the 672 plots associated with 2015 year, FAMily8 experiment.

Plot table:

The plot table contains all of the information about plots that have been grown in the field. This is the crucial information for statistical analysis such as the replicate, sub-block, and row column. Also there is information about the entry and planting direction and pass, which could be useful for identify mistakes in the field and unraveling unexpected patterns in the field.

Plot_id: This is the primary key for the table and the most important column. This links the plot to phenotypic information. **To make your plot_id.** Each plot gets a unique ID that is composed of three parts concatenated together:

1. Two digit year.
2. Three letter location or experiment.
3. Five digit number.

Thus 15CIM00001 is from 2015 year, CIMmyt location, and plot 00001.

Source_seed_id: Where the seed to plant the trial came from. If it came from previous plot could have old plot_id or it could come from a selection.

Range: Planting range. ***Needed for statistical models if using row column design.**

Column: Planting column. ***Needed for statistical models if using row column design.**

Location: Location of the experiment.

Planting_date: Date the experiment was planted.

Person: Who is in charge of the experiment.

Rep: Planting replication. ***Needed for statistical models.**

Block: Planting block, often referred to as sub-block. ***Needed for statistical models.**

Experiment_id: Foreign key that links the experiment table to plot table. This allows one query to identify all plots that belong to an experiment.

Plot_name: Name of the entry such as a variety, line, etc.

Entry: Usually a numeric number assigned to the plot_name.

Purpose: Purpose of the plot, for example is it an entry or a check variety.

Treatment: The treatment that was imposed on the plots.

Pedigree: Pedigree of the plot_name or entry.

Serp_order: Serpentine order of plots.

Planting_side: For planting three row plots, which side of the planter was a plot assigned.

Planting_order: What the order was when planted.

Planting_direction: Return or go pass.

Planting_pass: What pass of planting the trial.

Notes: Additional information about the plots.

Queries based on the plot table:

1. Join with plot map to get plot boundaries:

```
Select plot.*, plot_map.*
From plot, plot_map
Where plot.plot_id=plot_map.plot_id and plot.experiment_id = '15_OBR_FAM8'
```

Returns 672 records with plot information and plot boundaries.

2. Identify all phenotypic data associated with the 15_ASH_BYD trials

```
Select phenotype.*
From phenotype, plot
Where phenotype.entity_id = plot.plot_id and plot.experiment_id = '15_ASH_BYD'
```

Returns 1654 phenotypic records of grain weight and moisture.

3. Identify all of the same information for grain weight as above query but join plot information for statistical analysis.

```
Select phenotype.*, plot.*
From phenotype, plot
Where phenotype.entity_id = plot.plot_id and plot.experiment_id = '15_ASH_BYD'
and phenotype.trait_id='GRWT'
```

Returns 827 records with plot information which can be easily imported for statistical analysis.

4. More complex query that returns all HTP data that is assigned to a particular plot (or experiment of plots) using a subquery.

```
Select phemu_htp.*
From phemu_htp
Where phemu_htp.plot_id in (select plot.plot_id from plot where plot.plot_id =
'16BYD00001')
```

Returns 108 records that are associated with plot 16BYD00001

5. Repeat query 4 only this time get all plots associated with 16BYD plot_id's

```
Select phemu_htp.*
```

From phemu_htp
Where phemu_htp.plot_id in (select plot.plot_id from plot where plot.plot_id like '16BYD%')

Returns 52839 records (as of September 10, 2016)

6. Find all plots associated with experiment table experiment_id.

Select plot.*
From plot, experiment
Where plot.experiment_id = experiment.experiment_id and experiment.experiment_id = '15_OBR_FAM8'

Returns 1344 records.

Phenotype table:

The phenotype table stores all the phenotypic data that was recorded.

Phenotype_id: Primary key, self-augmenting counter for each number.

Entity_id: Foreign key which is equivalent to plot_id in the plot table. This is essential to find data associated with plots.

Seed_selection: Was data taken on a single plant or bulk plot?

Trait_id: Phenotype trait measured. Foreign key to the trait_id table. Any trait that is recorded should have a valid trait_id.

Phenotype_value: Value that is observed. Can be numeric, character, date, etc. All dates should be entered as 2015-12-10.

Phenotype_date: Date that the phenotype was recorded.

Person: Person that recorded the phenotype.

Notes: Often used for seed selection.

Queries based on phenotype table:

1. Find all phenotype information related to the 15_ASH_BYD experiment.

Select phenotype.*
From phenotype
Where phenotype.entity_id in (select plot.plot_id from plot where plot.experiment_id = '15_ASH_BYD')

Returns 1654 records for grain weight and moisture content.

2. Pair phenotypic data with georeferenced plot data.

Select phenotype.*, plot_map.*
From phenotype, plot_map
Where plot_map.plot_id = phenotype.entity_id and phenotype.entity_id in (select plot.plot_id from plot where plot.experiment_id = '14_OBR_FAM8')

Returns 4047 georeferenced records.

3. Find all grain yield phenotype information related to the 15_ASH_BYD experiment.

Select phenotype.*
From phenotype
Where phenotype.entity_id in (select plot.plot_id from plot where plot.experiment_id = '15_ASH_BYD') and phenotype.trait_id = 'GRWT'

Returns 827 records.

Traits table:

The traits table provides the key to decode what the phenotype trait_id is known as. Each phenotype entered should have a valid trait_id filled out in the trait table. Most data collected should have already known traits, thus there is little to upload to the table, but it is important to know what information is stored within.

Trait: 3-5 character string that abbreviates the trait. This is what is entered in the trait_id of the phenotype table.

Format: Is value an numeric, text, categorical, date, etc.

DefaultValue: Value of the trait assumed by default.

Minimum: Minimum value that a trait can assume.

Maximum: Maximum value that a trait can assume.

Details: Information about the trait.

Categories: For a categorical trait, the categories that are recognized.

Ontology_id: Ontology of the trait as accepted by Crop Science and other databases. (For wheat traits, refer to http://www.cropontology.org/ontology/CO_321/Wheat)

Trait_name: Full trait name.

Full_description: Trait description such as plant height is measured from soil level to top of the spike without measuring the awns.

High-Throughput Phenotyping Data:

High-throughput phenotyping (HTP) refers to collecting massive amounts of geo-referenced data using land and aerial systems. Currently, the Poland lab is using land-based systems like the PheMU and Phenocart along with aerial systems of small unmanned aerial vehicles (UAVs). While each platform is unique in capabilities and measurements, post-processing pipelines have been developed that curate and store much of the data with minimal user support. For introductory information and how to utilize tables in the database the following overview is provided. **NOTE:** This is not an extensive listing of the columns in each table and only the PheMU tables have been highlighted. Other HTP system tables are similar and by highlighting the PheMU tables the users should be able to translate that information to other systems.

Plot_map table:

This table provides the georeferenced coordinates for each plot using Universal Transverse Mercator (UTM) coordinates. This table should be filled in once for each experiment, and will be used to identify data that is associated with each plot through spatial queries.

Record_id: Primary key for the table that is an auto-incremented.

Plot_id: Same as plot table plot_id, allows user to join plots with spatial information.

Plot_polygon: Plot coordinates entered into a format to make GIS polygons. Useful for spatial queries using the spatial plugin.

C1_1_x: UTM coordinates in the x (east-west direction) for coordinate 1.

C1_1_y: UTM coordinates in the y (north-south direction) for coordinate 1.

C1_2_x: UTM coordinates in the x (east-west direction) for coordinate 2.

C1_2_y: UTM coordinates in the y (north-south direction) for coordinate 2.

C2_1_x: UTM coordinates in the x (east-west direction) for coordinate 3.

C2_1_y: UTM coordinates in the y (north-south direction) for coordinate 3.

C2_2_x: UTM coordinates in the x (east-west direction) for coordinate 4.

C2_2_y: UTM coordinates in the y (north-south direction) for coordinate 4.

Together the 8 columns define 4 (x,y) point pairs that make a polygon around the plot. Standard polygon is below; however, it is quite possible that the naming convention is not consistent across all plots.

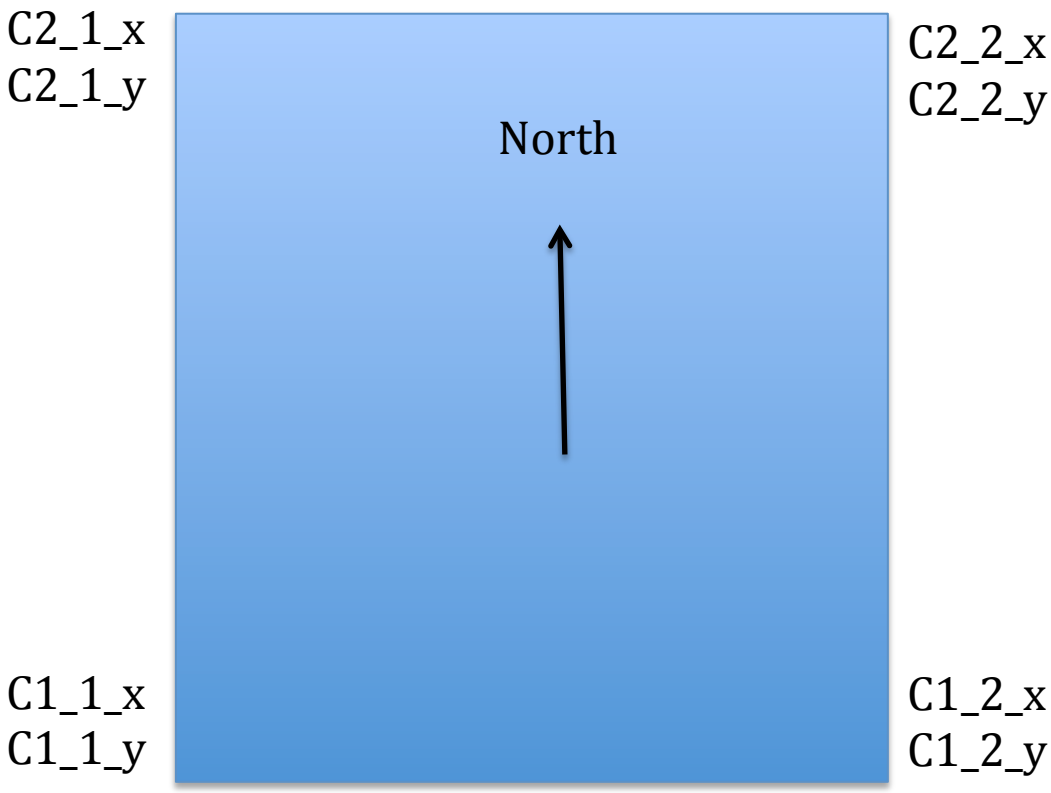

Figure 1. Example of a field plot with the four (x, y) coordinate pairs defining the plot.

Queries utilizing the plot_map table:

1. Query plots assuming that the plot is a box that aligns north and south. This query is deprecated with the implementation of spatial plugins.

```
SELECT  h.* , p.*, pid.*
FROM htp  h inner join
plot_map p on
p.C1_1_x>h.absolute_sensor_position_x and p.C1_2_x<h.absolute_sensor_position_x
and p.C2_1_y>h.absolute_sensor_position_y and
p.C1_1_y<h.absolute_sensor_position_y
join plot pid on p.plot_id = pid.plot_id
WHERE pid.plot_name = 'roelf'  and pid.purpose = 'canopy_structure'
```

Returns 7340 records that include plot information, HTP, data, and plot map information

Phemu_run table:

This table provides information about each run of a particular phenotyping platform. Each system has its own table but information about the runs are similar. The run table provides a high level over view of all the HTP data that has been collected for a

particular phenotyping system. **The table is auto-populated from the HTP processing pipelines, users should not need to change any information in this table.** Key information about this table are:

Run_id: Unique run_id that is give to each data set. It is formed by concatenating the starting date, starting time, ending date, and ending time with "_." Dates and times are expressed as 4 digit year, 2 digit month, 2 digit day (20161029 for October 29, 2016) and 24 hour, minute, and second for time.

Start_data_utc: Starting date of the phenotyper run, given by 4 digit year, 2 digit month, and 2 digit year using coordinated universal time (UTC).

Start_time_utc: Time of start of phenotype run given by 2 digit hour on a 24 hour clock, 2 digit minute, and 2 digit second time. All time is stored as UTC time.

End_date_utc: Ending date formed as start date.

End_time_utc: Ending time formed similar to start time.

Run_folder_name: Each run of raw data is stored in its own unique folder. This is the location of the raw data.

Minimum and maximum latitude and longitude: Locations of the maximum and minimum latitude and longitude within the file.

Location, region, and country: Experiment location given by area.

Notes: Often include the experiment_id.

Phemu_images table:

This table provides identifying location for each image that was taken using the phenotyping system. Much like the run table, this table is auto-populated by the HTP processing pipeline. **NOTE: Users should not directly manipulate any data on this table.** Columns that are most likely of interest to the user are:

Image_file_name: Unique file name for each image. Made up of camera_sn number, sampling_date, and then the picture number given from the camera.

Run_id: Links to the run table which would allow users to identify where data is stored.

Camera_sn: Serial number for each camera. Used to make part of the image file name.

Plot_id: Plot_id that the image is assigned to. Plot assigning occurs from automated script, and the plot_id can be used to query phenotype and plot information.

Absolute_sensor_position_(x and y): Position where the image was taken.

Sampling_time: Sampling time given in UTC, number is hours:minutes:seconds.

Sampling_date: Sampling date given by year-month-date.

Queries using the image table:

1. Identify all images that are associated with plot_id "15ASH20685."

```
Select phemu_images.*
From phemu_images
Where phemu_images.plot_id = "15ASH20685"
```

Returns 251 records.

2. Spatial query to populate and return plot_id for all rows associated with a specific year (2016) and a specific run_id (phemu_20160422_160839_20160422_175823):

```
Select phemu_images.*
From phemu_images INNER JOIN plot_map on
ST_CONTAINS(plot_map.plot_polygon, phemu_images.position)
Where plot_map.plot_id LIKE "16%" AND phemu_images.run_id LIKE
"phemu_20160422_160839_20160422_175823"
```

Returns 27,292 records of images.

Phemu_htp table:

This table stores all sensor data that is not photos. The information in this table is automated from pipelines processing the raw phenotypic data. **NOTE: Users should not directly manipulate any data on this table.** Ultimately, this table records the value of the sensor reading, date, time, position, and sensor type. It allows for the sensor to be offset from the GPS unit and tracks both the location of the GPS receiver and where the sensor is in relation to the sensor. Absolute_sensor_position_(x and y) account for any sensor offset and is the location of the sensor in space. Columns of most interest are:

Run_id: The run_id for the phenotyping platform. This links to the run table.

Plot_id: The plot_id where the data is located. This is automatically assigned using the plot and plot map tables. Data that is not assigned to a plot maintains a NULL value. This column provides a way to query data for certain plots.

Sensor_id: The sensor_id of the sensors. Relates to the htp_instrument table.

Sensor_observation: Value provided by the sensor.

Absolute_sensor_position_(x and y): Two columns that together form an x-y pair that locates the sensor observation in space.

Sampling_time_utc: Sampling time provided in UTC time.

Sampling_date_utc: Sampling date provided as UTC format.

Queries based on the htp table:

1. Identify all HTP data associated with plot "16BYD00341."

```
Select phemu_htp.*
From phemu_htp
Where phemu_htp.plot_id = "16BYD00341"
```

Returns 742 records.

2. Identify HTP data that is from a GreenSeeker sensor within plot "16BYD00341."

```
Select phemu_htp.*
From phemu_htp
Where phemu_htp.plot_id = "16BYD00341" and phemu_htp.sensor_id like 'GSK%'
```

Returns 340 records.

3. Identify HTP data for the 15_ASH_BYD experiment.

```
Select phemu_htp.*
From phemu_htp
Where phemu_htp.plot_id in (select plot.plot_id from plot where plot.experiment_id
= '15_ASH_BYD')
```

Returns 174,406 records (approximately 30 second run time)

4. Identify HTP data for 15_ASH_BYD experiment and infrared (temperature) sensors.

```
Select phemu_htp.*
From phemu_htp
Where phemu_htp.plot_id in (select plot.plot_id from plot where plot.experiment_id
= '15_ASH_BYD') and phemu_htp.sensor_id like 'IRT%'
```

Returns 52,752 records.

5. Identify HTP data across multiple years and locations that is associated with BYD.

```
Select phemu_htp.*
From phemu_htp
Where phemu_htp.plot_id in (select plot.plot_id from plot where plot.experiment_id
like '%BYD%')
```

Returns 513,488 records which have both 2015 and 2015 BYD plot information.

6. Spatial query to populate plot_id for all rows associated with a specific year (2016), a specific run_id (phemu_20160422_160839_20160422_175823), and plot_id (16ASH00001):

```
Select phemu_images.*
From phemu_images INNER JOIN plot_map on
ST_CONTAINS(plot_map.plot_polygon, phemu_images.position)
Where plot_map.plot_id LIKE "16%" AND phemu_images.run_id LIKE
"phemu_20160422_160839_20160422_175823" and plot_map.plot_id =
"16ASH00001"
```

Returns 35 records of images associated with plot 16ASH00001.

Htp_instrument table:

This table stores information about each sensor that is used in the Poland lab including what phenotyping the platform is located on. Abbreviated table columns are:

Sensor_id: Sensor_id is made of 3-5 letters describing the sensor (GSK=GreenSeeker, CCL=Crop Circle, IRT=infrared thermometer, USC=ultrasonic sensor) and the sensor serial number.

Sensor_model: Type of sensor.

Serial_number: Sensor serial number used to make sensor_id.

Platform: The HTP platform the sensor is used on.

Notes: Other manufacture information about sensor specifications.

Genomic Data (GBS and DNA):

The genotyping tables provide a method for users to keep track of DNA samples from collection, quantification, to sequencing. The following tables provide information about the data that is stored and how it is curated.

GBS table:

The GBS table is essential for making GBS key files and documenting the plexing, lanes, and flowcell that were used for sequencing. It also provides a high level over-view of genomic data that has been collected. Columns are:

Gbs_id: Unique ID for each GBS run. Formed from concatenating GBS and 5 digit number.

GBS_name: Name for GBS plates. Should be informative.

Dna_id: Plate_id for the DNA plates that contain the samples. Links to the DNA table. The DNA_ID is made from the date that the plate was made followed by the number of plate made for that particular day.

Flowcell: Flowcell identification for sequencing. Essential for making GBS Key File.

Lane: Lane that was sequenced for particular GBS library. Essential for making GBS Key File.

Plexing: Sample plexing level. Links to the barcodes table. Essential for making GBS Key File.

Project: Identification for the project, often with larger projects that have many plates one project ID groups multiple plates together.

Person: Person who is charge of the GBS project.

Enzyme: Enzyme used for the GBS library.

Species: Plant or animal species of the DNA.

Avg_size: Average size of the DNA submitted for sequencing.

Notes: Often used for more description about the library contents.

Library_date: Day the library was made.

Submitted_date: Date the GBS library was submitted to sequencing facility.

Facility: Location that sequenced the GBS library.

Run_date: Date that the sequencing run was started (usually not used.)

Run_id: Sequencing run ID (usually not used.)

Order_num: Purchase order number for the sequencing services

Account_num: Account to be charged if PO is not used to order sequencing services.

Platform: Type of sequencing machine.

Num_lines: Number of lines in the sequencing file.

md5sum: MD5SUM for the sequencing file.

Global:

Invoice: Indicates whether an invoice needs to be generated in order to charge for the sequencing services.

Queries using the GBS table:

1. Find all plates related to the Karl Ventnor population.

```
Select gbs.*
From gbs
Where gbs.project = 'KarlVentnor'
```

Returns 2 records.

2. Find all with gbs_name of 'DusterBillings.'

```
Select gbs.*
From gbs
Where gbs.gbs_name like '%Duster%'
```

Returns 6 records.

DNA table:

The DNA table stores all of the DNA sample plate information, including sample name and location on the DNA plate. This table is linked to the gbs table through the plate_id. Columns are:

Sample_id: Unique sample ID made up of "DNA" followed by the date the DNA plate was made, plate number, and then the well of the plate occupied by the sample. Year is 2 digit number, example—DNA160825P03_G07 is DNA plate made August 25, 2016, plate number 3 well G07.

Plate_id: Unique plate_id that is linked to the GBS table. Plate_id is made by forming DNA with the date and plate number. Plate_id—DNA160825P03 is DNA plate made August 25, 2016, plate number 3.

Plate_name: Descriptive name for the plate. Often the project and plate number.

Well_A01: Well the sample occupies written with letter followed by number H10.

Well_01A: Well the sample is in written by the well number and then letter, the reciprocal of well_A01, 10H.

Sample_name: Name of the sample. **This is critical to make the GBS key as it provides the FullSampleName**.

Tissue_id: Information about the tissue used for DNA extraction.

External_id: Column that addition information could be recorded, especially if the germplasm is from a different program such as CIMMYT.

IMPORTANT: The **Sample_name, Tissue_id,** and **External_id** columns have been used differently by various experimenters, thus there may be differences from plate to plate. It is possible that **Sample_name** may be blank and **Tissue_id** or **External_id** may be used to form the FullSampleName for the GBS key.

Tissue_type: Type of tissue that was collected, leaf, root, seed.

Extraction: Method used for DNA extraction.

Dna_person: Person who extracted the DNA.

Notes: Additional information.

Date: Date the DNA extraction was completed.

Line_num: Usually not used.

Species: Species of the plant or animal that DNA was extracted.

Queries using the DNA table:

1. Make a GBS key file for Family 8 population for GBSv5 TASSEL pipeline.

```
SELECT gbs.flowcell Flowcell,
gbs.lane Lane,
barcodes.barcode Barcode,
dna.sample_name FullSampleName,
dna.plate_id,
```

```
substring(dna.well_A01,1,1),
substring(dna.well_A01,2,2),
concat(gbs.gbs_id,barcodes.barcode),
dna.well_A01,
dna.notes,
gbs.plexing,
gbs.project,
gbs.enzyme,
dna.sample_id,
dna.tissue_id,
dna.external_id,
dna.dna_person,
dna.line_num,
gbs.gbs_name,
dna.plate_name,
dna.well_01A,
plant.plant_name,
gbs.gbs_id,
barcodes.`set`
FROM dna LEFT JOIN gbs ON gbs.dna_id = dna.plate_id
LEFT JOIN plant ON dna.tissue_id = plant.plant_id
INNER JOIN barcodes ON dna.well_A01 = barcodes.well_A01 AND
gbs.plexing LIKE barcodes.`set1
WHERE gbs.project LIKE "%crain%"
and gbs.`global` = 1
ORDER BY gbs.gbs_id, dna.well_01A ASC
```

Returns 1536 records that serve as a key to run the GBS TASSEL pipeline.

DNAQuantification table:

This table stores the DNA quantification values for DNA plates. The sample, sample, values, and location are saved for each plate. **Note: Most of the information is auto-completed for this table.** Columns of interest are:

Sample_id: Linked to DNA table sample_id column. Include "q" plus DNA sample_id.

Plate_name: Organization or project for the plat.

Quant_id:

Flor_val: Fluorescence value of measured sample.

Quant_val: Quantity value of DNA for each sample

Quant_method: Method used to quantify DNA.

Quant_person: Who performed the DNA quantification.

Date: Date that quantification occurred.

Where to go from here:

This guide has provided a quick over view of the most common tables and queries that many users will need. Future documentation will provide more detail about each table. Mark Lucas, mlucas@ksu.edu, is also the database administrator and can assist with data questions.