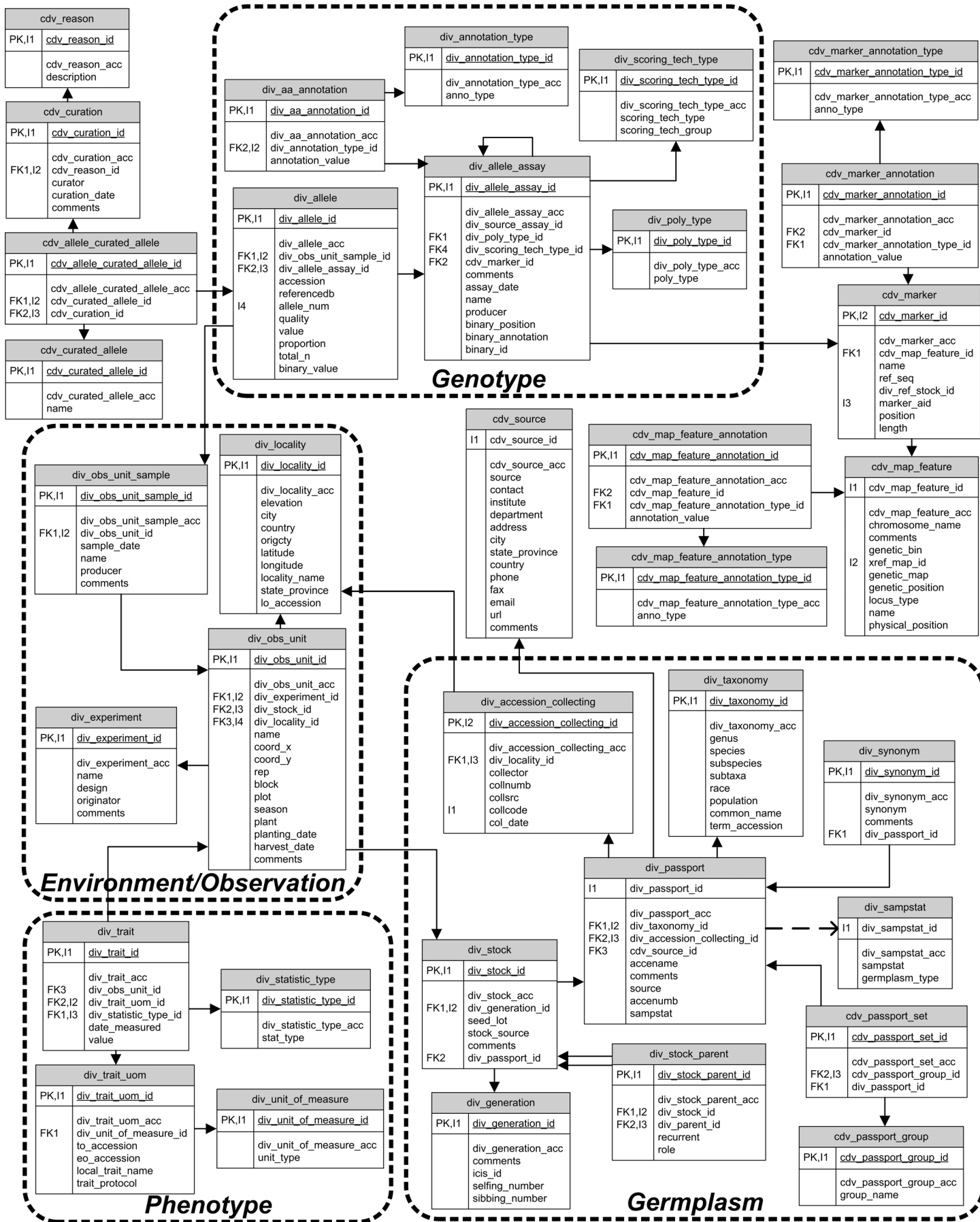


# The Genomic Diversity and Phenotype Data Model (GDPDM) – Version 4.1 February 19, 2010



**Table/Field descriptions for  
The Genomic Diversity and Phenotype Data Model (GDPDM)  
Version 4.1 February 19, 2010**

<b>Table/Field</b>	<b>Type</b>	<b>Req</b>	<b>Description</b>
<b>div_aa_annotation</b>	table		This table is used to annotate an allele assay. Any given allele assay can have multiple annotations. For example, this table is used to describe the primers used, apparatus used to score the marker, or the nature of the repeat in an SSR. Please only store annotations associated with assays.
div_aa_annotation_id	int	yes	primary key
div_aa_annotation_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_aa_annotation:435.
div_annotation_type_id	int	yes	foreign key referencing the annotation type.
div_allele_assay_id	int	yes	foreign key referencing the allele assay this is annotating.
annotation_value	text	yes	value for given type of this annotation.
<b>div_accession_collecting</b>	table		This table stores information about where, when, and by whom an accession was collected or released. This information is expected only for germplasm collected in the wild or from landraces. This is not required for material produced by plant breeders.
div_accession_collecting_id	int	yes	primary key
div_accession_collecting_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_accession_collecting:435.
div_locality_id	int	no	foreign key to locality
collector	varchar	no	Collector of the accession
collnumb	varchar	yes	Collecting number: Original number assigned by the collector(s) of the sample, normally composed of the name or initials of the collector(s) followed by a number. This number is essential for identifying duplicates held in different collections.
collsrc	varchar	no	Collecting/acquisition source: The coding scheme proposed can be used at 2 different levels of detail: either by using the general codes (in boldface) such as 10, 20, 30, 40 or by using the more specific codes such as 11, 12 etc. 10) Wild habitat 11) Forest/woodland 12) Shrubland 13) Grassland 14) Desert/tundra 15) Aquatic habitat 20) Farm or cultivated habitat 21) Field 22) Orchard 23) Backyard, kitchen or home garden (urban, peri-urban or rural) 24) Fallow land 25) Pasture 26) Farm store 27) Threshing floor 28) Park 30) Market or shop 40) Institute, Experimental station, Research organization, Genebank 50) Seed company 60) Weedy, disturbed or ruderal habitat 61) Roadside 62) Field margin 99) Other (Elaborate in REMARKS field).

Table/Field	Type	Req	Description
collcode	varchar	no	Collecting institute code: Code of the Institute collecting the sample. If the holding institute has collected the material, the collecting institute code (COLLCODE) should be the same as the holding institute code (INSTCODE). Follows INSTCODE standard.
col_date	date	no	Date of this collection or release.
<b>div_allele</b>	table		This table stores the genotypic or allele data. A separate record is made for each allele. THIS TABLE HOLDS THE ACTUAL MOLECULAR DATA. If the assay can differentiate all three classes (i.e. not dominant), then there should be two alleles (records). All the SSR and SNP data that we have should have two alleles and two records.
div_allele_id	int	yes	primary key
div_allele_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_allele:435.
div_obs_unit_sample_id	int	yes	foreign key referencing observational unit sample.
div_allele_assay_id	int	yes	foreign key referencing allele assay.
allele_num	int	yes	number used to uniquely identify this allele from other alleles with the same observational unit sample and allele assay.
quality	text	no	Quality values should be computed using the equation: $Q = -10 \log_{10}(\text{Probability of error})$ . Explanation from phrap documentation: Phred computes a probability p of an error in the base call at each position, and converts this to a quality value q using the transformation $q = -10 \log_{10}(p)$ . Thus a quality of 30 corresponds to an error probability of 1 / 1000, a quality of 40 to an error probability of 1 / 10000, etc. The quality value of 99 should be reserved for base calls that have been visually inspected and verified as "highly accurate" (during editing), and 98 for bases that have been edited but are not highly accurate (these are converted to quality 0 in phrap).
value	text	yes	Value of the allele. Examples include: For a sequence-type polymorphism, the value would be the actual sequence. For an SSR-type polymorphism, the value would be the actual length in bp of this allele. For a null allele enter string "NULL_ALLELE". For allele not detected enter "NOT_DETECTED". A null allele with low confidence would be recorded as "NOT_DETECTED". Any no reaction would not be stored in database.
proportion	double	no	number of alleles with given value divided by total_n
total_n	int	no	total number of homologous chromosomes surveyed (for diploids it is the number of gametes)
accession	varchar	no	The accession designation provided by an international sequence or SNP repository. For example: NCBI Genbank or dbSTS accessions of the DNA sequences.
referencedb	varchar	no	The name of the international sequence or SNP repository.
binary_value	medium blob	no	Byte 1-3: Bit 1-24: Version of the compression algorithm (3 bytes) (ASCII). This version: 001. All data will be stored in big-endian format.
			Byte 4: Bit 25-32: Blob Type (1 byte) (ASCII). This blob: 1. There should be one zip entry per type of BLOB.

Table/Field	Type	Req	Description
			Byte 5-8: Bit 33-64: Number of Sites (SNP/Indel positions) (4 bytes)
			Byte 9-18: Bit 65-144: Genome Version (10 byte) (ASCII)
			Byte 19-43: Bit 145-344: Chromosome Name (25 bytes) (ASCII)
			Byte 44-47: Bit 345-376: Start Position (4 bytes). Smallest Absolute Physical Position.
			Byte 48-51: Bit 377-408: End Position (4 bytes). Largest Absolute Physical Position
			Byte 52-201: Accession Name (150 bytes) (ASCII)
			Remaining bytes through byte 1024 are unused for unknown header information. All unused bytes will be null bytes (0x00).
			Byte 1025 - end: Each SNP/Indel genotyping value uses 4 Bits, represented by IUPAC nucleotide code. Code '-' represents Indels. Code 'N' will be unknown. Values are smallest physical position to largest. If there are an odd number of sites, 4 bits (0x0) will be added to the end.
	<b>hex</b>		<b>snp value</b>
	0x0	A	A
	0x1	C	C
	0x2	G	G
	0x3	T	T
	0x4	R	A or G
	0x5	Y	C or T
	0x6	S	G or C
	0x7	W	A or T
	0x8	K	G or T
	0x9	M	A or C
	0xA	B	C or G or T
	0xB	D	A or G or T
	0xC	H	A or C or T
	0xD	V	A or C or G
	0xE	N	any base
	0xF	-	gap
<b>div_allele_assay</b>	table		This table records information on HOW, WHEN, AND BY WHOM a particular molecular or genotypic assay was conducted. The assay may be a SNP assay, a PCR-direct sequencing assay, etc. Typically, genotyping experiments are made of multiple assays. In addition, there are multiple assays that can be used to score a given marker. For example, SSR or sequencing technologies could both be used to score a single marker, therefore there would be two assays for the same marker.
div_allele_assay_id	int	yes	primary key
div_allele_assay_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_allele_assay:435.
cdv_marker_id	int	no	foreign key referencing the marker associated with the reference sequence.
div_source_assay_id	int	no	foreign key referencing source assay within this same table.

Table/Field	Type	Req	Description
div_poly_type_id	int	yes	foreign key referencing the polymorphism type of this assay
div_scoring_tech_type_id	int	no	foreign key referencing its scoring technology type.
comments	text	no	comments about this allele assay.
assay_date	date	yes	date of this allele assay.
name	varchar	yes	name of this assay. For Seq. Alignments: "MarkerName Sequence Alignment" For other assays: "MarkerName ScoringTechnology Assay" Example: "bnlg1014 SSR Assay"
producer	varchar	no	producer's name of this assay. (i.e. name of lab)
length	int	no	Deprecated: Use cdv_marker.length instead.
position	varchar	no	Deprecated: Use cdv_marker.position instead.
ref_seq	text	yes	Deprecated: Use cdv_marker.ref_seq instead.
div_ref_stock_id	int	no	Deprecated: Use cdv_marker.div_ref_stock_id instead.
binary_position	medium blob	no	Byte 1-3: Bit 1-24: Version of the compression algorithm (3 bytes) (ASCII). This version: 001. All data will be stored in big-endian format.
			Byte 4: Bit 25-32: Blob Type (1 byte) (ASCII). This blob: 2. There should be one zip entry per type of BLOB.
			Byte 5-8: Bit 33-64: Number of Sites (SNP/Indel positions) (4 bytes)
			Byte 9-18: Bit 65-144: Genome Version (10 byte) (ASCII)
			Byte 19-43: Bit 145-344: Chromosome Name (25 bytes) (ASCII)
			Byte 44-47: Bit 345-376: Start Position (4 bytes). Smallest Absolute Physical Position.
			Byte 48-51: Bit 377-408: End Position (4 bytes). Largest Absolute Physical Position
			Byte 52-201: NULL (150 bytes)
			Remaining bytes through byte 1024 are unused for unknown header information. All unused bytes will be null bytes (0x00).
			Byte 1025 - end: SNP/Indel absolute physical positions from smallest to largest. Each position is a 4 byte integer.
binary_annotation	medium blob	no	Byte 1-3: Bit 1-24: Version of the compression algorithm (3 bytes) (ASCII). This version: 001. All data will be stored in big-endian format.
			Byte 4: Bit 25-32: Blob Type (1 byte) (ASCII). This blob: 6. There should be one zip entry per type of BLOB.
			Byte 5-8: Bit 33-64: Number of Sites (SNP/Indel positions) (4 bytes)
			Byte 9-18: Bit 65-144: Genome Version (10 byte) (ASCII)
			Byte 19-43: Bit 145-344: Chromosome Name (25 bytes) (ASCII)
			Byte 44-47: Bit 345-376: Start Position (4 bytes). Smallest Absolute Physical Position.
			Byte 48-51: Bit 377-408: End Position (4 bytes). Largest Absolute Physical Position
			Byte 52-201: NULL (150 bytes)
			Remaining bytes through byte 1024 are unused for unknown header information. All unused bytes will be null bytes (0x00).
			Byte 1025 - end: Sizes of Indels (4 bytes each). Should be in the same order as SNP/Indel values/positions above. Here SNP values have size 0.

Table/Field	Type	Req	Description
binary_id	medium blob	no	Byte 1-3: Bit 1-24: Version of the compression algorithm (3 bytes) (ASCII). This version: 001. All data will be stored in big-endian format.
			Byte 4: Bit 25-32: Blob Type (1 byte) (ASCII). This blob: 5. There should be one zip entry per type of BLOB.
			Byte 5-8: Bit 33-64: Number of Sites (SNP/Indel positions) (4 bytes)
			Byte 9-18: Bit 65-144: Genome Version (10 byte) (ASCII)
			Byte 19-43: Bit 145-344: Chromosome Name (25 bytes) (ASCII)
			Byte 44-47: Bit 345-376: Start Position (4 bytes). Smallest Absolute Physical Position.
			Byte 48-51: Bit 377-408: End Position (4 bytes). Largest Absolute Physical Position
			Byte 52-201: NULL (150 bytes)
			Byte 202-205: Bit 1609-1640: SNP ID Length (4 bytes)
			Remaining bytes through byte 1024 are unused for unknown header information. All unused bytes will be null bytes (0x00).
			Byte 1025 - end: Each entry (length defined by bytes 202-205) contains SNP ID in same order as SNPs in binary_value field.
<b>div_annotation_type</b>	table		This table provides the controlled vocabulary for annotating allele assays. For example, PRIMER1, PRIMER2, SEQUENCING_DYE, ALLELE_REPEAT, CONTIG_PROGRAM, ALIGNMENT_ANNOTATION, CLONING_PROCESS are currently annotation categories that are recorded.
div_annotation_type_id	int	yes	primary key
div_annotation_type_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_annotation_type:435.
anno_type	varchar	yes	value of this annotation type. Below are standardized anno_type values.
primer1			5' primer used for PCR amplification or sequencing
primer2			3' primer used for PCR amplification or sequencing
sequencing_dye			dye used in sequencing reaction, eg. "ABI Big Dye"
allele_repeat			Sequence motif of the repeat for an SSR marker (e.g., "CCG", "AG")
contig_program			program used for making contigs, eg. "Biolign"
alignment_annotation			comments for sequence alignments
cloning_process			Method of cloning. eg: "Direct PCR"
source_genbank			Source Genbank accession
source_contig_name			Source Contig Name
tracking_name			Tracking Name
allele_repeat_size			Size of repeat for an SSR marker ("di", "tri", "tetra" etc.)
<b>div_experiment</b>	table		table of experiments. An experiment is a collection of observational units with one originator that are part of a specific study. An experiment may be conducted at more than one location and in more than one season or year.
div_experiment_id	int	yes	primary key

Table/Field	Type	Req	Description
div_experiment_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_experiment:435.
name	varchar	yes	name of this experiment
design	varchar	no	design of this experiment
originator	varchar	yes	originator of this experiment
comments	text	no	comments
<b>div_generation</b>	table		table of generation values. A generation value describes the number and type of generations that separate a taxon from an original source cross or population as defined in the div_stock_parent table. For example, F2, F7, or BC2F4.
div_generation_id	int	yes	primary key
div_generation_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_generation:435.
value	varchar	yes	Deprecated: Use selfing_number and sibbing_number. generation value
comments	text	no	text description
icis_id	text	no	reference to ICIS generation entity
selfing_number	int	no	The number of generations of self pollination separating the parental cross from the stock.
sibbing_number	int	no	The number of generations of full-sib crossing separating the parental cross from the stock.
<b>div_locality</b>	table		table used to store localities
div_locality_id	int	yes	primary key
div_locality_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_locality:435.
elevation	int	no	Elevation of collecting site: [m asl] Elevation of collecting site expressed in metres above sea level. Negative values are allowed.
city	varchar	no	city
country	varchar	no	country in plain text
origcty	varchar	no	Country of origin: Code of the country in which the sample was originally collected. Use the 3-letter ISO 3166-1 extended country codes.
latitude	double	no	Latitude of collecting site: this should be recorded as a decimal number. North latitude should be positive values and South latitude should be negative values.
longitude	double	no	Longitude of collecting site: this should be recorded as a decimal number. West longitude should be positive values and East longitude should be negative values.
locality_name	varchar	no	name of this locality
state_province	varchar	no	state or province
lo_accession	varchar	no	Gazetteer Ontology Term ID.

Table/Field	Type	Req	Description
<p>Gazetteer Ontology (GAZ) is an important reference source for information about places and place-names and includes information on the geographical makeup of a continent, country, region, state/province, district, reGENCY, subdistrict, sub-reGENCY, city, waterways, lakes, rivers, mountains etc. IT IS EXPECTED THAT THE CURATOR FILLS IN THE ID OF THE FINEST OR THE EXACT LOCALITY E.G. ITHACA AND REST OF THE INFO ON COUNTY/STATE/COUNTRY, ETC. IS FILLED BY QUERYING THE GAZ. IF YOU THINK THE GAZ TERM(S) REQUIRED BY YOU IS MISSING, PLEASE WRITE TO GRAMENE@GRAMENE.ORG. You can search/browse the existing terms in the ontology by visiting <a href="http://www.gramene.org/plant_ontology">http://www.gramene.org/plant_ontology</a>.</p>			
<b>div_obs_unit</b>	table		table of observational units. An observational unit is a plant or group of plants on which trait measurements are recorded or from which DNA samples are derived. Typically, it is a plot in a field but can also be a plant or pot in a greenhouse or growth chamber.
div_obs_unit_id	int	yes	primary key
div_obs_unit_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_obs_unit:435.
div_experiment_id	int	no	foreign key referencing associated experiment
div_stock_id	int	yes	foreign key referencing stock planted
div_locality_id	int	yes	foreign key referencing associated locality
name	varchar	yes	name of observational unit
coord_x	int	no	x coordinate
coord_y	int	no	y coordinate
rep	varchar	no	group of blocks
block	varchar	no	group of plots
plot	varchar	no	smallest measure observed. (i.e. row in field)
season	varchar	no	season
plant	varchar	no	Arbitrary plant number (i.e. 1, 2, 3, pool).
planting_date	date	no	date planted (example format: 4/19/2004)
harvest_date	date	no	date harvested (example format: 4/19/2004)
comments	text	no	comments
<b>div_obs_unit_sample</b>	table		This table defines the genotypic or molecular sample taken from a particular observation unit. Typically, this is the DNA prep from a particular plant.
div_obs_unit_sample_id	int	yes	primary key
div_obs_unit_sample_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_obs_unit_sample:435.
div_obs_unit_id	int	yes	foreign key referencing observational unit this sample was collected from
sample_date	date	yes	date sample collected
name	varchar	yes	name of this sample
producer	varchar	yes	producer of this sample
comments	text	no	comments
<b>div_passport</b>	table		table of taxa. Each record in this table represents a distinct genetic entity, which is expected to be represented by one or more physical stocks. NOTE: Many of the passport, taxonomy, and accession_collecting fields were derived from the FAO/IPGRI Multi-Crop Passport Descriptors ( <a href="http://www.ipgri.cgiar.org/publications/pdf/124.pdf">http://www.ipgri.cgiar.org/publications/pdf/124.pdf</a> )

Table/Field	Type	Req	Description
div_passport_id	int	yes	primary key
div_passport_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_passport:435.
div_taxonomy_id	int	no	foreign key to div_taxonomy table
div_accession_collecting_id	int	no	foreign key to div_accession_collecting table
cdv_source_id	int	no	foreign key to cdv_source table.
accename	varchar	yes	Accession name: Either a registered or other formal designation given to the accession. First letter uppercase. Alternate names should be entered in the synonym table.
comments	text	no	Remarks: The remarks field is used to add notes or to elaborate on descriptors with value 99 or 999 (=Other). Prefix remarks with the field name they refer to and a colon (e.g. COLLSRC:roadside). Separate remarks referring to different fields are separated by semicolons without space.
germplasm_type	varchar	no	Deprecated: Use sampstat instead. Germplasm type of accession (i.e. Tripsacum, inbred)
source	varchar	yes	Institute code or person's name. Immediate source of germplasm. Use cdv_source to store contact information.
accenumb	varchar	no	Accession number: Actually an ASCII string, which serves as the unique identifier for accessions within a genebank collection, and is assigned when a sample is entered into the genebank collection.
sampstat	varchar	no	Biological status of accession: The coding scheme proposed can be used at 3 different levels of detail: either by using the general codes such as 100, 200, 300, 400 or by using the more specific codes such as 110, 120 etc. 100) Wild 110) Natural 114) Inbred line developed from wild material. 120) Semi-natural/wild 200) Weedy 300) Traditional cultivar/landrace 314) Inbred line developed from a landrace 400) Breeding/research material 410) Breeder's line 411) Synthetic population 412) Hybrid 413) Founder stock/base population 414) Inbred line (parent of hybrid cultivar) 415) Segregating population 420) Mutant/genetic stock 500) Advanced/improved cultivar 999) Other (Elaborate in REMARKS field).
div_poly_type	table		table used to store/control possible polymorphism types. This provides information so that analysis can understand what type of data is stored in div_allele.value. SNP=(A,C, G, T, or N); LENGTH = Number (generally from SSR, AFLP, or RFLP study); SEQUENCE=aligned IUPAC sequence; CATEGORICAL=(eg. A, B, or H; +,-)
div_poly_type_id	int	yes	primary key
div_poly_type_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_poly_type:435.
poly_type	varchar	yes	value of polymorphism (SNP, LENGTH, SEQUENCE, CATEGORICAL).
div_sampstat	table		This table translates the sampstat codes into friendly strings.

Table/Field	Type	Req	Description
div_sampstat_id	int	yes	primary key
div_sampstat_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_sampstat:435.
sampstat	varchar	yes	sampstat code. Same as in div_passport.
germplasm_type	varchar	yes	User friendly string for given sampstat code. Same as the deprecated field in div_passport.
<b>div_scoring_tech_type</b>	table		The technology used to generate the data for a specific allele assay.
div_scoring_tech_type_id	int	yes	primary key
div_scoring_tech_type_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_scoring_tech_type:435.
scoring_tech_group	varchar	yes	The field holds the larger grouping of the scoring technology. (eg. RFLP, SSR, CAPS, AFLP, SNP, Sequencing, Unknown)
scoring_tech_type	varchar	yes	Holds a detailed description of the specific technology used (eg. SNP detection by single base extension and mass spec. detection).
<b>div_statistic_type</b>	table		table used to store/control possible statistic types
div_statistic_type_id	int	yes	primary key
div_statistic_type_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_statistic_type:435.
stat_type	varchar	yes	value of statistic type (measure, mean, variance)
<b>div_stock</b>	table		table of stocks. A stock is the specific source of a genetic entity from which an observational unit is taken. For most species, a stock will be a packet of seed in cold storage. An established line or cultivar is represented as a passport while a specific seedlot of that line is represented as a stock.
div_stock_id	int	yes	primary key
div_stock_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_stock:435.
div_generation_id	int	no	foreign key referencing stock's generation
div_passport_id	int	yes	foreign key referencing stock's passport information
seed_lot	varchar	yes	an alphanumeric id that can be used by the source to access a specific stock and information about it. The combination of source and seedlot will be unique for a stock.
stock_source	varchar	no	the individual or institution that PROVIDED the stock directly used to create a specific obs_unit [the person may no longer maintain or have it, i.e., if the entire stock was used in the experiment]
comments	text	no	comments

Table/Field	Type	Req	Description
<b>div_stock_parent</b>	table		table used to define the parents of the cross or population from which a stock is derived
div_stock_parent_id	int	yes	primary key
div_stock_parent_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_stock_parent:435.
div_stock_id	int	yes	foreign key referencing child stock
div_parent_id	int	yes	foreign key referencing parent stock (the stock_id of the parent)
recurrent	int	yes	Integer value indicating number of times this parent repeatedly crossed. Default value is 0. Example: A = ((B x C) x B) x B. Parent B would have recurrent value of 2. Parent C would have recurrent value 0.
role	varchar	yes	Value indicating role of this parent (i.e. self, male, female).
<b>div_synonym</b>	table		Synonyms or alternate names for passport accession names (passport.accname)
div_synonym_id	int	yes	primary key
div_synonym_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_synonym:435.
div_passport_id	int	yes	Foreign key referencing div_passport table.
synonym	varchar	yes	The synonym.
comments	text	no	Notes about the synonym.
<b>div_taxonomy</b>	table		table used to store taxonomy
div_taxonomy_id	int	yes	primary key
div_taxonomy_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_taxonomy:435.
genus	varchar	no	Genus: Genus name for taxon. Initial uppercase letter required.
species	varchar	yes	Species: Specific epithet portion of the scientific name in lowercase letters. Following abbreviation is allowed: 'sp.'
subspecies	varchar	no	Subspecies
subtaxa	varchar	no	Subtaxa: Subtaxa can be used to store any additional taxonomic identifier. Following abbreviations are allowed: 'convar.' (for convariety); 'var.' (for variety); 'f.' (for form).
race	varchar	no	
population	varchar	no	Name of the population
common_name	varchar	no	Common name of the crop (i.e. Wild Rice).
term_accession	varchar	no	Gramene Taxonomy Ontology term ID
<p>Gramene Taxonomy Ontology (GR_tax) is an important reference source for the taxonomy tree in the ontology format. Each term in this ontology can represent varietas, subspecies, species, genus, order, class or any rank in the classification. IT IS EXPECTED THAT THE CURATOR FILLS IN THE ID OF THE MOST GRANULAR TAXON. This taxonomy ontology focuses on the Poaceae (Gramineae) family of plant taxonomy only. IF YOU THINK THE TAXONOMY TERM(S) REQUIRED BY YOU IS MISSING, PLEASE WRITE TO GRAMENE@GRAMENE.ORG. You can search/browse the existing terms in the ontology by visiting <a href="http://www.gramene.org/plant_ontology">http://www.gramene.org/plant_ontology</a>.</p>			

Table/Field	Type	Req	Description
<b>div_trait</b>	table		table used to store trait values for an observational unit. THIS IS WHERE THE REAL PHENOTYPIC (TRAIT) AND TREATMENT DATA GOES.
div_trait_id	int	yes	primary key
div_trait_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_trait:435.
div_trait_uom_id	int	yes	foreign key referencing trait being measured
div_statistic_type_id	int	yes	foreign key referencing statistic type
div_obs_unit_id	int	yes	foreign key referencing observational unit
date_measured	date	yes	date trait measured
value	varchar	yes	value of trait
<b>div_trait_uom</b>	table		table used to control trait names, units of measure, and a link to the Gramene Trait Ontology.
div_trait_uom_id	int	yes	primary key
div_trait_uom_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_trait_uom:435.
div_unit_of_measure_id	int	yes	foreign key referencing unit of measure
to_accession	varchar	no	Trait Ontology term ID
<p>Trait Ontology (TO) is an important reference source for the qualitative and quantitative traits (e.e. leaf color, plant height, days to flowering) assayed for phenotyping in the ontology format. IT IS EXPECTED THAT THE CURATOR FILLS IN THE ID OF THE MOST GRANULAR TAXON. IF YOU THINK THE TAXONOMY TERM(S) REQUIRED BY YOU IS MISSING, PLEASE WRITE TO GRAMENE@GRAMENE.ORG. You can search/browse the existing terms in the ontology by visiting <a href="http://www.gramene.org/plant_ontology">http://www.gramene.org/plant_ontology</a>.</p>			
eo_accession	varchar	no	Plant Environment Ontology term ID
<p>Plant Environment Ontology (EO) is an important reference source for the environments and treatments (e.g. watering regime, IAA, ABA, UV-A) supplemented to evaluate qualitative and quantitative traits assayed for phenotyping. IT IS EXPECTED THAT THE CURATOR FILLS IN THE ID OF THE MOST GRANULAR TAXON. IF YOU THINK THE TAXONOMY TERM(S) REQUIRED BY YOU IS MISSING, PLEASE WRITE TO GRAMENE@GRAMENE.ORG. You can search/browse the existing terms in the ontology by visiting <a href="http://www.gramene.org/plant_ontology">http://www.gramene.org/plant_ontology</a>.</p>			
local_trait_name	varchar	yes	this defines the trait being measured and should correspond with the qtl_trait_ontology_id. This will always be defined even when a qtl_trait_ontology_id for the measured trait does not exist.
trait_protocol	text	no	Very important. A thorough description of how the trait was collected, and if a rating, the minimum and maximum values
<b>div_unit_of_measure</b>	table		used to store/control possible units of measure
div_unit_of_measure_id	int	yes	primary key
div_unit_of_measure_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:div_unit_of_measure:435.
unit_type	varchar	yes	defines unit of measure. (i.e. inches, lbs, etc)
<b>cdv_allele_curated_allele</b>	table		Many-to-many between div_allele, cdv_curated_allele, and cdv_curation
cdv_allele_curated_allele_id	int	yes	primary key

Table/Field	Type	Req	Description
cdv_allele_curated_allele_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:cdv_allele_curated_allele:435.
cdv_curated_allele_id	int	yes	foreign key to cdv_curated_allele
div_allele_id	int	yes	foreign key to div_allele
cdv_curation_id	int	yes	foreign key to cdv_curation: Who assigned this observation to this curated allele?
<b>cdv_curated_allele</b>	table		Curated allele name corresponding to observed div_alleles
cdv_curated_allele_id	int	yes	Primary key
cdv_curated_allele_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:cdv_curated_allele:435.
name	varchar	yes	Curated name for this allele.
<b>cdv_curation</b>	table		General curation table.
cdv_curation_id	int	yes	Primary key
cdv_curation_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:cdv_curation:435.
cdv_reason_id	int	yes	Reason for an allele grouping, e.g., same SSR length. Foreign key to cdv_reason
curator	varchar	yes	Name of curator
curation_date	date	yes	Date curated
comments	text	no	Comments
<b>cdv_map_feature</b>	table		This table stores information on the location of the marker within the genome. Since genomic mapping evolves rapidly in unsequenced genomes, this information is expected to change rapidly, unlike the marker and assay data, which should be constant once deposited. The cdv_marker.ref_seq data can be used to map markers in sequenced genomes.
cdv_map_feature_id	int	yes	Primary key
cdv_map_feature_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:cdv_map_feature:435.
chromosome_name	text	no	Chromosome name where this locus resides.
comments	text	no	Comments
genetic_bin	varchar	no	Position of the locus within a bin
genetic_map	varchar	no	Name of genetic map (map set)
xref_map_id	varchar	no	External database reference to a genetic map (e.g., "cu-dh-2001-6")
genetic_position	double	no	Position in cM along a chromosome
locus_type	varchar	yes	Type of this locus (i.e. Gene, Cytological).
name	varchar	yes	Name of this locus.
physical_position	double	no	Physical position of this locus.

Table/Field	Type	Req	Description
<b>cdv_map_feature_annotation</b>	table		This table is used to annotate a cdv_map_feature. Any given cdv_map_feature can have multiple annotations. For example, describe the reference sequence, Genbank Accession, genetic map position.
cdv_map_feature_annotation_id	int	yes	Primary key
cdv_map_feature_annotation_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:cdv_map_feature_annotation:435.
cdv_map_feature_id	int	yes	Foreign key to cdv_map_feature
cdv_map_feature_annotation_type_id	int	yes	Foreign key to cdv_map_feature_annotation_type
annotation_value	text	yes	value for given type of this annotation.
<b>cdv_map_feature_annotation_type</b>	table		This table provides the controlled vocabulary for annotating cdv_map_features. For example, Ref_seq, Ref_seq_div_allele_id are currently annotation categories that are recorded.
cdv_map_feature_annotation_type_id	int	yes	Primary key
cdv_map_feature_annotation_type_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:cdv_map_feature_annotation_type:435.
anno_type	varchar	yes	value of this annotation type. Below are standardized anno_type values.
ref_seq_cDNA			Representative cDNA sequence of the locus.
ref_seq			The representative genomic sequence of the locus. In maize database, it is the sequence from B73 if available.
<map name>_chr			Suggested naming convention for additional chromosome mappings.
<b>cdv_marker</b>	table		This table defines the marker being scored. Markers are meiotically defined alleles that exist at a single locus. Typically, this will be a SNP, sequence repeat, or a sequence region that is polymorphic. In some cases, such as AFLP the sequence will not be known, but the band can be assigned name. This table does NOT describe the reagents used to score the marker (use div_allele_assay for that).
cdv_marker_id	int	yes	Primary key
cdv_marker_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:cdv_marker:435.
cdv_map_feature_id	int	yes	Foreign key to cdv_map_feature
div_ref_stock_id	int	no	foreign key referencing stock associated with the reference sequence.
length	int	no	length of polymorphism being assayed relative to reference sequence.
position	int	no	position of polymorphism being assayed relative to reference sequence.
Examples: 1) A SNP would have a position=1 and length=1; 2) A sequence assay might have position 1 and 500bp of length (length of the alignment); 3) A SSR would have a position and particular length in the reference.; 4) A Affy hybridization scoring reaction would have position 1 and length of 25.			
marker_aid	varchar	no	reference into Gramene database
name	varchar	yes	name

Table/Field	Type	Req	Description
ref_seq	text	no	reference sequence
			The reference sequence is typically annotated, such that the scored polymorphism is placed in brackets (eg. []). If the whole sequence is being scored, as in a sequencing experiment then the whole sequence is bracketed. Suggested convention: SNP polymorphisms will be given in alphabetical order -- eg. [A/G] or {C/T} rather than [G/A] or {T/C}. Indels will be indicated by [TCG/] or {CGTA/----}.
<b>cdv_marker_annotation</b>	table		This table is used to annotate a cdv_marker. Any given cdv_marker can have multiple annotations. Please only store annotations for markers.
cdv_marker_annotation_id	int	yes	Primary key
cdv_marker_annotation_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:cdv_marker_annotation:435.
cdv_marker_id	int	yes	Foreign key to cdv_marker
cdv_marker_annotation_type_id	int	yes	Foreign key to cdv_marker_annotation_type
annotation_value	text	yes	value for given type of this annotation.
<b>cdv_marker_annotation_type</b>	table		This table provides the controlled vocabulary for annotating cdv_map_features. For example, Ref_seq, Ref_seq_div_allele_id are currently annotation categories that are recorded.
cdv_marker_annotation_type_id	int	yes	Primary key
cdv_marker_annotation_type_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:cdv_marker_annotation_type:435.
anno_type	varchar	yes	value of this annotation type. Below are standardized anno_type values.
marker_type			Marker type (SNP, Sequencing, SSR, INDEL, CAPS, etc). Used when no assay has been done.
original_name			The original name of the marker -- usually the marker name used in the contributing laboratory
primer1			5' primer used for PCR amplification or sequencing
primer2			3' primer used for PCR amplification or sequencing
allele_repeat			Sequence motif of the repeat for an SSR marker (e.g., "CCG", "AG")
allele_repeat_size			Size of repeat for an SSR marker ("di", "tri", "tetra" etc.)
left_tag			The 6 characters to the left of a SNP marker, plus the SNP itself. eg. ACTACG[C/T]
right_tag			The 6 characters to the right of a SNP marker, plus the SNP itself. eg. [C/T]AGCTAC
<b>cdv_passport_group</b>	table		A user-selected working set of germplasm lines.
cdv_passport_group_id	int	yes	Primary key
cdv_passport_group_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:cdv_passport_group:435.
name	varchar	no	User's name for this set of lines.
<b>cdv_passport_set</b>	table		Many/many relation table for cdv_passport_group to div_passport

Table/Field	Type	Req	Description
cdv_passport_set_id	int	yes	Primary key
cdv_passport_set_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:cdv_passport_set:435.
cdv_passport_group_id	int	yes	Foreign key
div_passport_id	int	yes	Foreign key
<b>cdv_reason</b>	table		Controlled vocabulary of reasons for creating a curation entry.
cdv_reason_id	int	yes	Primary key
cdv_reason_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:cdv_reason:435.
description	text	yes	E.g., Same SSR length
<b>cdv_source</b>	table		Contact and identification information for seed/plant providers.
cdv_source_id	int	yes	Primary key
cdv_source_acc	text	no	Unique global key used across multiple databases. The keys will be assigned with the following format: <authority>:<table name>:<object id> For example: Gramene:cdv_source:435.
source	varchar	yes	Name/Identifier of the source. Full text name of source.
contact	varchar	no	Contact person.
institute	varchar	no	Institute
department	varchar	no	Department
address	varchar	no	Address
city	varchar	no	City
state_province	varchar	no	State/Province
country	varchar	no	Country
phone	varchar	no	Phone number
fax	varchar	no	Fax number
email	varchar	no	E-mail
url	text	no	URL
comments	text	no	Comments