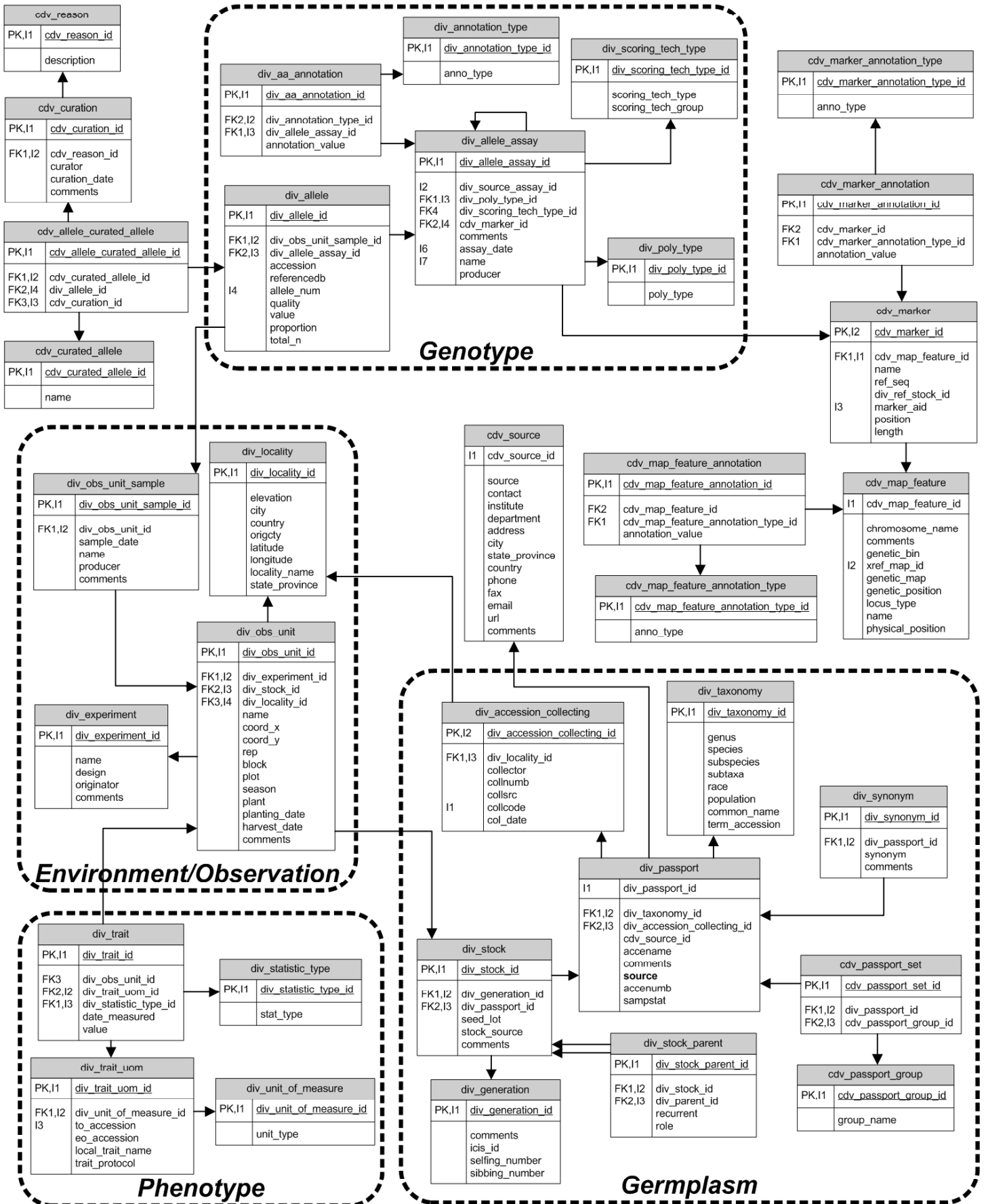


The Genomic Diversity and Phenotype Data Model (GDPDM) - Version 2.1 April 4, 2006



**Table/Field descriptions for
The Genomic Diversity and Phenotype Data Model (GDPDM)
Version 2.1 November 20, 2006**

Table/Field	Type	Req	Description
div_aa_annotation	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_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.
div_accession_collecting	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_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).
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.

Table/Field	Type	Req'	Description
div_allele	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_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	int	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.
referencedb	varchar	no	The name of the international sequence or SNP repository.
div_allele_assay	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
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.
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.

Table/Field	Type	Req	Description
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.
div_annotation_type	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
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.)
div_experiment	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
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
div_generation	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
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.

Table/Field	Type	Req'	Description
div_locality	table		table used to store localities
div_locality_id	int	yes	primary key
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
div_obs_unit	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_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
div_obs_unit_sample	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_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

Table/Field	Type	Req	Description
div_passport	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 (http://www.ipgri.cgiar.org/publications/pdf/124.pdf)
div_passport_id	int	yes	primary key
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
poly_type	varchar	yes	value of polymorphism (SNP, LENGTH, SEQUENCE, CATEGORICAL).
div_scoring_tech_type	table		The technology used to generate the data for a specific allele assay.
div_scoring_tech_type_id	int	yes	primary key
scoring_tech_group	varchar	yes	The field holds the larger grouping of the scoring technology. (eg. RFLP, SSR, CAPS, AFLP, SNP, Sequencing, Unknown)

Table/Field	Type	Req'	Description
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).
div_statistic_type	table		table used to store/control possible statistic types
div_statistic_type_id	int	yes	primary key
stat_type	varchar	yes	value of statistic type (measure, mean, variance)
div_stock	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_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
div_stock_parent	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_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).
div_synonym	table		Synonyms or alternate names for passport accession names (passport.accname)
div_synonym_id	int	yes	primary key
div_passport_id	int	yes	Foreign key referencing div_passport table.
synonym	varchar	yes	The synonym.
comments	text	no	Notes about the synonym.
div_taxonomy	table		table used to store taxonomy
div_taxonomy_id	int	yes	primary key
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	

Table/Field	Type	Req'	Description
population	varchar	no	Name of the population
common_name	varchar	no	Common name of the crop (i.e. Wild Rice).
term_accession	varchar	no	taxonomy ontology accession. Used to reference external taxonomy ontology.
div_trait	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_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
div_trait_uom	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_unit_of_measure_id	int	yes	foreign key referencing unit of measure
to_accession	varchar	no	foreign key referencing trait being measured (reference into Gramene database).
eo_accession	varchar	no	foreign key referencing type of treatment being applied (reference into Gramene database).
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
div_treatment	table		<i>Deprecated: Use div_trait.</i> table used to define treatment values or levels
div_treatment_id	int	yes	primary key
div_obs_unit_id	int	yes	foreign key referencing observational unit being treated
div_treatment_uom_id	int	yes	foreign key referencing type of treatment being applied
value	varchar	yes	value of this treatment
div_treatment_uom	table		<i>Deprecated: Use div_trait_uom.</i> table used to control treatment units of measure and descriptors. Treatments can be experimental treatments or environment variables.
div_treatment_uom_id	int	yes	primary key
eo_accession	varchar	yes	foreign key referencing type of treatment being applied (reference into Gramene database).
div_unit_of_measure_id	int	yes	foreign key referencing unit of measure
local_treatment_name	varchar	yes	local treatment name
treatment_protocol	varchar	no	comments
div_unit_of_measure	table		used to store/control possible units of measure
div_unit_of_measure_id	int	yes	primary key
unit_type	varchar	yes	defines unit of measure. (i.e. inches, lbs, etc)
cdv_allele_curated_allele	table		Many-to-many between div_allele, cdv_curated_allele, and cdv_curation
cdv_allele_curated_allele_id	int	yes	primary key
cdv_curated_allele_id	int	yes	foreign key to cdv_curated_allele

Table/Field	Type	Req'	Description
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?
cdv_curated_allele	table		Curated allele name corresponding to observed div_alleles
cdv_curated_allele_id	int	yes	Primary key
name	varchar	yes	Curated name for this allele.
cdv_curation	table		General curation table.
cdv_curation_id	int	yes	Primary key
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
cdv_map_feature	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
chromosome_name	text	yes	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.
cdv_map_feature_annotation	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_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.
cdv_map_feature_annotation_type	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
anno_type	varchar	yes	value of this annotation type. Below are standardized anno_type values.
ref_seq			The representative genomic sequence of the locus. In maize database, it is the sequence from B73 if available.

Table/Field	Type	Req'	Description
cdv_marker	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_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
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/} rather than by [TCG/-] or {CGTA/----}.
cdv_marker_annotation	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_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.
cdv_marker_annotation_type	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
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

Table/Field	Type	Req	Description
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
cdv_passport_group	table		A user-selected working set of germplasm lines.
cdv_passport_group_id	int	yes	Primary key
name	varchar	no	User's name for this set of lines.
cdv_passport_set	table		Many/many relation table for cdv_passport_group to div_passport
cdv_passport_set_id	int	yes	Primary key
cdv_passport_group_id	int	yes	Foreign key
div_passport_id	int	yes	Foreign key
cdv_reason	table		Controlled vocabulary of reasons for creating a curation entry.
cdv_reason_id	int	yes	Primary key
description	text	yes	E.g., Same SSR length
cdv_source	table		Contact and identification information for seed/plant providers.
cdv_source_id	int	yes	Primary key
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