Section 5

SQL: Data Definition Language

Data Definition Language

- Create
  - Table
  - Index
  - View
- Alter
  - Table
- Drop
  - Table
  - Index
  - View
- Grant
- Revoke

The following slides use MySQL syntax.

Create Database

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
    [create_specification] ...

create_specification:
    [DEFAULT] CHARACTER SET [=] charset_name
  | [DEFAULT] COLLATE [=] collation_name
```

`CREATE DATABASE` creates a database with the given name. A database in MySQL is implemented as a directory containing files that correspond to tables in the database. Because there are no tables in a database when it is initially created, the `CREATE DATABASE` statement creates only a directory under the MySQL data directory and the `db.opt` file.

Create Database

```
paul@charis:~/410/mysql_docker/mysql-data$ ll
total 188460
drwxr-xr-x 7  999 docker     4096 Oct 11 23:44 ./
drwxrwxr-x 5 paul paul       4096 Sep 25 14:48 ../
-rw-r----- 1  999 docker       56 Sep 17 10:23 auto.cnf
-rw-r----- 1  999 docker     1033 Oct 11 23:43 ib_buffer_pool
-rw-r----- 1  999 docker 79691776 Oct 11 23:44 ibdata1
-rw-r----- 1  999 docker 50331648 Oct 11 23:45 ib_logfile0
-rw-r----- 1  999 docker 50331648 Sep 17 10:23 ib_logfile1
-rw-r----- 1  999 docker 12582912 Oct 16 17:46 ibtmp1
drwxr-x--- 2  999 docker     4096 Sep 26 12:06 learning_sql/
drwxr-x--- 2  999 docker     4096 Sep 17 10:23 mysql/
drwxr-x--- 2  999 docker     4096 Sep 17 10:23 performance_schema/
drwxr-x--- 2  999 docker    12288 Sep 17 10:23 sys/
drwxr-x--- 2  999 docker     4096 Sep 26 12:07 uncle_ted_sample/
```

Create Database

```
paul@charis:~/410/mysql_docker/mysql-data/learning_sql$ ll
total 11500
drwxr-x--- 2 999 docker     4096 Sep 26 12:06 ./
drwxr-xr-x 7 999 docker     4096 Oct 11 23:44 ../
-rw-r----- 1 999 docker     9041 Sep 26 12:06 account.frm
-rw-r----- 1 999 docker   163840 Sep 26 12:06 account.ibd
-rw-r----- 1 999 docker     8726 Sep 26 12:06 branch.frm
-rw-r----- 1 999 docker    98304 Sep 26 12:06 branch.ibd
-rw-r----- 1 999 docker     8678 Sep 26 12:06 business.frm
-rw-r----- 1 999 docker    98304 Sep 26 12:06 business.ibd
-rw-r----- 1 999 docker     8794 Sep 26 12:06 customer.frm
-rw-r----- 1 999 docker    98304 Sep 26 12:06 customer.ibd
-rw-r----- 1 999 docker       65 Sep 26 12:06 db.opt
-rw-r----- 1 999 docker     8596 Sep 26 12:06 department.frm
-rw-r----- 1 999 docker    98304 Sep 26 12:06 department.ibd
-rw-r----- 1 999 docker     8886 Sep 26 12:06 employee.frm
-rw-r----- 1 999 docker   147456 Sep 26 12:06 employee.ibd
-rw-r----- 1 999 docker     8672 Sep 26 12:06 individual.frm
-rw-r----- 1 999 docker    98304 Sep 26 12:06 individual.ibd
-rw-r----- 1 999 docker     8784 Sep 26 12:06 officer.frm
-rw-r----- 1 999 docker   114688 Sep 26 12:06 officer.ibd
-rw-r----- 1 999 docker     8746 Sep 26 12:06 product.frm
-rw-r----- 1 999 docker   114688 Sep 26 12:06 product.ibd
-rw-r----- 1 999 docker     8612 Sep 26 12:06 product_type.frm
-rw-r----- 1 999 docker    98304 Sep 26 12:06 product_type.ibd
-rw-r----- 1 999 docker     8894 Sep 26 12:06 transaction.frm
-rw-r----- 1 999 docker 10485760 Oct 10 08:23 transaction.ibd
```

Create Database

```
paul@charis:~/410/mysql_docker/mysql-data/learning_sql$ cat db.opt
default-character-set=latin1
default-collation=latin1_swedish_ci
```

Create Table (Syntax 1)

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    (create_definition,...)
    [table_options]
    [partition_options]
```

Create Table (Syntax 2)

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    [(create_definition,...)]
    [table_options]
    [partition_options]
    [IGNORE | REPLACE]
    [AS] query_expression
```

Create Table (Syntax 3)

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
    { LIKE old_tbl_name | (LIKE old_tbl_name) }
```

Create Table Syntax

```
create_definition:
    col_name column_definition
  | [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (index_col_name,...)
      [index_option] ...
  | {INDEX|KEY} [index_name] [index_type] (index_col_name,...)
      [index_option] ...
  | [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY]
      [index_name] [index_type] (index_col_name,...)
      [index_option] …
  | [CONSTRAINT [symbol]] FOREIGN KEY
      [index_name] (index_col_name,...) reference_definition
  | CHECK (expr)
```

(abbreviated)

Create Table Syntax 1

```
column_definition:
    data_type [NOT NULL | NULL] [DEFAULT default_value]
      [AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]
      [COMMENT 'string']
      [reference_definition]
  | data_type [GENERATED ALWAYS] AS (expression)
      [VIRTUAL | STORED] [UNIQUE [KEY]] [COMMENT comment]
      [NOT NULL | NULL] [[PRIMARY] KEY]
```

Create Table Syntax 1

NULL vs DEFAULT NULL vs NULL DEFAULT NULL in MYSQL column creation?

There is no difference. `NULL DEFAULT NULL` is the implicit default.

From the CREATE TABLE documentation:

- If neither NULL nor NOT NULL is specified, the column is treated as though NULL had been specified.

From the "Data Type Default Values" chapter:

- If a column definition includes no explicit DEFAULT value, MySQL determines the default value as follows: If the column can take NULL as a value, the column is defined with an explicit DEFAULT NULL clause.

http://stackoverflow.com/a/25968248

MySQL Data Types (Numeric)

```
data_type:
    BIT[(length)]
  | TINYINT[(length)] [UNSIGNED] [ZEROFILL]
  | SMALLINT[(length)] [UNSIGNED] [ZEROFILL]
  | MEDIUMINT[(length)] [UNSIGNED] [ZEROFILL]
  | INT[(length)] [UNSIGNED] [ZEROFILL]
  | INTEGER[(length)] [UNSIGNED] [ZEROFILL]
  | BIGINT[(length)] [UNSIGNED] [ZEROFILL]
  | REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]
  | DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
  | FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]
  | DECIMAL[(length[,decimals])] [UNSIGNED] [ZEROFILL]
  | NUMERIC[(length[,decimals])] [UNSIGNED] [ZEROFILL]
```

MySQL Data Types (Temporal)

```
data_type:
    | DATE
    | TIME[(fsp)]
    | TIMESTAMP[(fsp)]
    | DATETIME[(fsp)]
    | YEAR
```

MySQL Data Types (String)

```
data_type:
  | CHAR[(length)] [BINARY]
      [CHARACTER SET charset_name] [COLLATE collation_name]
  | VARCHAR(length) [BINARY]
      [CHARACTER SET charset_name] [COLLATE collation_name]
```

MySQL Data Types

```
data_type:
    | BINARY[(length)]
    | VARBINARY(length)
    | TINYBLOB
    | BLOB
    | MEDIUMBLOB
    | LONGBLOB
```

MySQL Data Types

```
data_type:
  | TINYTEXT [BINARY]
      [CHARACTER SET charset_name] [COLLATE collation_name]
  | TEXT [BINARY]
      [CHARACTER SET charset_name] [COLLATE collation_name]
  | MEDIUMTEXT [BINARY]
      [CHARACTER SET charset_name] [COLLATE collation_name]
  | LONGTEXT [BINARY]
      [CHARACTER SET charset_name] [COLLATE collation_name]
```

MySQL Data Types

```
data_type:
  | ENUM(value1,value2,value3,...)
      [CHARACTER SET charset_name] [COLLATE collation_name]
  | SET(value1,value2,value3,...)
      [CHARACTER SET charset_name] [COLLATE collation_name]
  | JSON
  | spatial_type
```

MySQL Data Types

```
table_options:
    table_option [[,] table_option] ...
```

```
table_option:
    ENGINE [=] engine_name
  | AUTO_INCREMENT [=] value
  | AVG_ROW_LENGTH [=] value
  | [DEFAULT] CHARACTER SET [=] charset_name
  | CHECKSUM [=] {0 | 1}
  | [DEFAULT] COLLATE [=] collation_name
  | COMMENT [=] 'string'
  | COMPRESSION [=] {'ZLIB'|'LZ4'|'NONE'}
  | CONNECTION [=] 'connect_string'
  | DATA DIRECTORY [=] 'absolute path to directory'
  | DELAY_KEY_WRITE [=] {0 | 1}
  | ENCRYPTION [=] {'Y' | 'N'}
  | INDEX DIRECTORY [=] 'absolute path to directory'
  | INSERT_METHOD [=] { NO | FIRST | LAST }
  | KEY_BLOCK_SIZE [=] value
  | MAX_ROWS [=] value
  | MIN_ROWS [=] value
  | PACK_KEYS [=] {0 | 1 | DEFAULT}
  | PASSWORD [=] 'string'
  | ROW_FORMAT [=] {DEFAULT|DYNAMIC|FIXED|COMPRESSED|REDUNDANT|COMPACT}
  | STATS_AUTO_RECALC [=] {DEFAULT|0|1}
  | STATS_PERSISTENT [=] {DEFAULT|0|1}
  | STATS_SAMPLE_PAGES [=] value
  | TABLESPACE tablespace_name [STORAGE {DISK|MEMORY|DEFAULT}]
  | UNION [=] (tbl_name[,tbl_name]...)
```

MySQL Data Types

```
reference_definition:
    REFERENCES tbl_name (index_col_name,...)
      [MATCH FULL | MATCH PARTIAL | MATCH SIMPLE]
      [ON DELETE reference_option]
      [ON UPDATE reference_option]

reference_option:
    RESTRICT | CASCADE | SET NULL | NO ACTION
```

MySQL Data Types

```
partition_options:
    PARTITION BY
        { [LINEAR] HASH(expr)
        | [LINEAR] KEY [ALGORITHM={1|2}] (column_list)
        | RANGE{(expr) | COLUMNS(column_list)}
        | LIST{(expr) | COLUMNS(column_list)} }
    [PARTITIONS num]
    [SUBPARTITION BY
        { [LINEAR] HASH(expr)
        | [LINEAR] KEY [ALGORITHM={1|2}] (column_list) }
      [SUBPARTITIONS num]
    ]
    [(partition_definition [, partition_definition] ...)]
```

MySQL Data Types

```
partition_definition:
    PARTITION partition_name
        [VALUES
            {LESS THAN {(expr | value_list) | MAXVALUE}
            |
            IN (value_list)}]
        [[STORAGE] ENGINE [=] engine_name]
        [COMMENT [=] 'comment_text' ]
        [DATA DIRECTORY [=] 'data_dir']
        [INDEX DIRECTORY [=] 'index_dir']
        [MAX_ROWS [=] max_number_of_rows]
        [MIN_ROWS [=] min_number_of_rows]
        [TABLESPACE [=] tablespace_name]
        [(subpartition_definition [, subpartition_definition] ...)]
```

```
subpartition_definition:
    SUBPARTITION logical_name
        [[STORAGE] ENGINE [=] engine_name]
        [COMMENT [=] 'comment_text' ]
        [DATA DIRECTORY [=] 'data_dir']
        [INDEX DIRECTORY [=] 'index_dir']
        [MAX_ROWS [=] max_number_of_rows]
        [MIN_ROWS [=] min_number_of_rows]
        [TABLESPACE [=] tablespace_name]
```

Create Table

https://www.postgresql.org/docs/9.5/static/sql-createtable.html

We will be covering only a small subset of the possible CREATE TABLE options.

```
CREATE [ [ GLOBAL | LOCAL ] { TEMPORARY | TEMP } | UNLOGGED ] TABLE [ IF NOT EXISTS ] table_name ( [
  { column_name data_type [ COLLATE collation ] [ column_constraint [ ... ] ]
    | table_constraint
    | LIKE parent_table [ like_option ... ] }
    [, ... ]
] )
[ INHERITS ( parent_table [, ... ] ) ]
[ WITH ( storage_parameter [= value] [, ... ] ) | WITH OIDS | WITHOUT OIDS ]
[ ON COMMIT { PRESERVE ROWS | DELETE ROWS | DROP } ]
[ TABLESPACE tablespace ]

CREATE [ [ GLOBAL | LOCAL ] { TEMPORARY | TEMP } | UNLOGGED ] TABLE [ IF NOT EXISTS ] table_name
    OF type_name [ (
  { column_name WITH OPTIONS [ column_constraint [ ... ] ]
    | table_constraint }
    [, ... ]
) ]
[ WITH ( storage_parameter [= value] [, ... ] ) | WITH OIDS | WITHOUT OIDS ]
[ ON COMMIT { PRESERVE ROWS | DELETE ROWS | DROP } ]
[ TABLESPACE tablespace ]

where column_constraint is:

[ CONSTRAINT constraint_name ]
{ NOT NULL |
  NULL |
  CHECK ( expression ) |
  DEFAULT default_expr |
  UNIQUE index_parameters |
  PRIMARY KEY index_parameters |
  REFERENCES reftable [ ( refcolumn ) ] [ MATCH FULL | MATCH PARTIAL | MATCH SIMPLE ]
    [ ON DELETE action ] [ ON UPDATE action ] }
[ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]

and table_constraint is:

[ CONSTRAINT constraint_name ]
{ CHECK ( expression ) |
  UNIQUE ( column_name [, ... ] ) index_parameters |
  PRIMARY KEY ( column_name [, ... ] ) index_parameters |
  EXCLUDE [ USING index_method ] ( exclude_element WITH operator [, ... ] ) index_parameters [ WHERE ( predicate ) ] |
  FOREIGN KEY ( column_name [, ... ] ) REFERENCES reftable [ ( refcolumn [, ... ] ) ]
    [ MATCH FULL | MATCH PARTIAL | MATCH SIMPLE ] [ ON DELETE action ] [ ON UPDATE action ] }
[ DEFERRABLE | NOT DEFERRABLE ] [ INITIALLY DEFERRED | INITIALLY IMMEDIATE ]

and like_option is:

{ INCLUDING | EXCLUDING } { DEFAULTS | CONSTRAINTS | INDEXES | STORAGE | COMMENTS | ALL }

index_parameters in UNIQUE, PRIMARY KEY, and EXCLUDE constraints are:

[ WITH ( storage_parameter [= value] [, ... ] ) ]
[ USING INDEX TABLESPACE tablespace ]

exclude_element in an EXCLUDE constraint is:

{ column | ( expression ) } [ opclass ] [ ASC | DESC ] [ NULLS { FIRST | LAST } ]
```

Create Table

```
CREATE TABLE Department (
  ddept CHAR(3) PRIMARY KEY,
  dname VARCHAR(100) NOT NULL,
  dsalary_budget NUMERIC(15,2)
);
```

Create Table

```
CREATE TABLE Course (
  cdept CHAR(3),
  ccourse INT,
  cname VARCHAR(100) NOT NULL,

  PRIMARY KEY (cdept, ccourse),

  CONSTRAINT fk_crs_dpt
    FOREIGN KEY (cdept)
    REFERENCES Department (ddept)
);
```

Create Table

```
CREATE TABLE Faculty (
  fid INT PRIMARY KEY,
  flast VARCHAR(50),
  ffirst VARCHAR(50),
  fmi CHAR(3),
  fdept CHAR(3) NOT NULL,
  fsalary NUMERIC(10,2),
  fmgr_id INT,

  CONSTRAINT fk_fac_dpt
    FOREIGN KEY (fdept)
    REFERENCES Department (ddept),

  CONSTRAINT fk_fac_mgr
    FOREIGN KEY (fmgr_id)
    REFERENCES Faculty (fid)
);
```

Create Table

```
CREATE TABLE Section (
  scall INT PRIMARY KEY,
  sdept CHAR(3) NOT NULL,
  scourse INT NOT NULL,
  sid INT,

  CONSTRAINT fk_sec_fac
    FOREIGN KEY (sid)
    REFERENCES Faculty (fid),

  CONSTRAINT fk_sec_crs
    FOREIGN KEY (sdept,scourse)
    REFERENCES Course (cdept,ccourse)
);
```

Example 1

```
CREATE TEMPORARY TABLE UncleTedQ6Temp
  (PRIMARY KEY  (ddept, fid))
  AS (
    SELECT *
    FROM Department, Faculty
    WHERE ddept = fdept
);
```

```
mysql> DESCRIBE UncleTedQ6Temp;
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| ddept          | varchar(3)   | NO   | PRI | NULL    |       |
| dname          | varchar(100) | NO   |     | NULL    |       |
| dsalary_budget | decimal(15,2)| YES  |     | NULL    |       |
| fid            | int(11)      | NO   | PRI | NULL    |       |
| flast          | varchar(50)  | YES  |     | NULL    |       |
| ffirst         | varchar(50)  | YES  |     | NULL    |       |
| fmi            | char(1)      | YES  |     | NULL    |       |
| fdept          | varchar(3)   | NO   |     | NULL    |       |
| fsalary        | decimal(10,2)| YES  |     | NULL    |       |
| fmgr_id        | int(11)      | YES  |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
10 rows in set (0.00 sec)
```

Example 1

```
CREATE TEMPORARY TABLE UncleTedQ6Temp
  (PRIMARY KEY  (ddept, fid))
  AS (
    SELECT *
    FROM Department, Faculty
    WHERE ddept = fdept
);
```

```
mysql> SELECT * FROM UncleTedQ6Temp;
+-------+-----------------+----------------+-------+---------+--------+------+-------+----------+--------
-+
| ddept | dname           | dsalary_budget | fid   | flast   | ffirst | fmi  | fdept | fsalary  | fmgr_id
|
+-------+-----------------+----------------+-------+---------+--------+------+-------+----------+--------
-+
| ADM   | Administration  |      200000.00 | 52110 | Smith   | Alice  | NULL | ADM   | 82000.00 |    NULL
|
| BIO   | Biology         |       40000.00 | 21004 | Perry   | Bill   | S    | BIO   | 21800.00 |   31890
|
| BIO   | Biology         |       40000.00 | 31890 | Coulsen | Mary   | NULL | BIO   | 21400.00 |   52110
|
| CSI   | Computer Science |      180000.00 | 12058 | Borys   | Ted    | J    | CSI   | 48000.00 |   22321
|
| CSI   | Computer Science |      180000.00 | 22321 | Brady   | Kathy  | M    | CSI   | 63400.00 |   52110
|
| CSI   | Computer Science |      180000.00 | 32000 | delBene | Bill   | S    | CSI   | 63500.00 |   22321
```

Example 2

```
DROP TABLE IF EXISTS FacultyDeptsCopy;
CREATE TABLE FacultyDeptsCopy
  (fdept CHAR UNIQUE)
  AS SELECT fdept FROM uncle_ted_sample.Faculty;

ERROR 1062 (23000): Duplicate entry 'CSI' for key 'fdept'
```

Example 2

```
DROP TABLE IF EXISTS FacultyDeptsCopy;
CREATE TABLE FacultyDeptsCopy
  (fid INT, fdept CHAR(3) UNIQUE)
  IGNORE
  AS SELECT fid, fdept FROM uncle_ted_sample.Faculty;
```

```
mysql> SELECT * FROM FacultyDeptsCopy;
+-------+-------+
| fid   | fdept |
+-------+-------+
| 52110 | ADM   |
| 21004 | BIO   |
| 12058 | CSI   |
| 12206 | ENG   |
+-------+-------+
4 rows in set (0.00 sec)
```

Example 2

```
DROP TABLE IF EXISTS FacultyDeptsCopy;
CREATE TABLE FacultyDeptsCopy
  (fid INT, fdept CHAR(3) UNIQUE)
  REPLACE
  AS SELECT fid, fdept FROM uncle_ted_sample.Faculty;
```

```
mysql> SELECT * FROM FacultyDeptsCopy;
+-------+-------+
| fid   | fdept |
+-------+-------+
| 52110 | ADM   |
| 31890 | BIO   |
| 32000 | CSI   |
| 47862 | ENG   |
+-------+-------+
4 rows in set (0.00 sec)
```

ALTER TABLE

```
ALTER [IGNORE] TABLE tbl_name
    [alter_specification [, alter_specification] ...]
    [partition_options]
```

ALTER TABLE

```
alter_specification:
    table_options
  | ADD [COLUMN] col_name column_definition
        [FIRST | AFTER col_name ]
  | ADD [COLUMN] (col_name column_definition,...)
  | DROP [COLUMN] col_name
  | RENAME [TO|AS] new_tbl_name
```

ALTER TABLE

```
alter_specification:
    table_options
  | ADD [CONSTRAINT [symbol]] PRIMARY KEY
       [index_type] (index_col_name,...) [index_option] ...
  | ADD [CONSTRAINT [symbol]]
       UNIQUE [INDEX|KEY] [index_name]
       [index_type] (index_col_name,...) [index_option] ...
  | ADD [CONSTRAINT [symbol]]
       FOREIGN KEY [index_name] (index_col_name,...)
       reference_definition
```

ALTER TABLE

```
alter_specification:
    table_options
  | DROP PRIMARY KEY
  | DROP {INDEX|KEY} index_name
  | DROP FOREIGN KEY fk_symbol
  | DISABLE KEYS
  | ENABLE KEYS
  | RENAME {INDEX|KEY} old_index_name TO new_index_name
```

ALTER TABLE

```
alter_specification:
    table_options
  | ADD {INDEX|KEY} [index_name]
        [index_type] (index_col_name,...) [index_option] ...
  | ADD FULLTEXT [INDEX|KEY] [index_name]
        (index_col_name,...) [index_option] ...
  | ADD SPATIAL [INDEX|KEY] [index_name]
        (index_col_name,...) [index_option] ...
  | DROP {INDEX|KEY} index_name
  | RENAME {INDEX|KEY} old_index_name TO new_index_name
```

ALTER TABLE

```
CREATE TABLE Department (
  ddept CHAR(3),
  dname VARCHAR(100),
  dsalary_budget NUMERIC(15,2)
);
```

```
ALTER TABLE Department
  ADD PRIMARY KEY (ddept);

ALTER TABLE Department
  MODIFY dname VARCHAR(100) NOT NULL;
```

ALTER TABLE

```
CREATE TABLE Course (
  cdept CHAR(3),
  ccourse INT,
  cname VARCHAR(100)
);
```

```
ALTER TABLE Course
  ADD PRIMARY KEY (cdept, ccourse);

ALTER TABLE Course
  MODIFY cname VARCHAR(100) NOT NULL;

ALTER TABLE Course
  ADD CONSTRAINT fk_crs_dpt
  FOREIGN KEY (cdept)
  REFERENCES Department (ddept);
```

ALTER TABLE

```
ALTER TABLE Faculty
  ADD PRIMARY KEY (fid);

ALTER TABLE Faculty
  MODIFY fdept CHAR(3) NOT NULL;

ALTER TABLE Faculty
  ADD CONSTRAINT fk_fac_dpt
  FOREIGN KEY (fdept)
  REFERENCES Department (ddept);

ALTER TABLE Faculty
  ADD CONSTRAINT fk_fac_mgr
  FOREIGN KEY (fmgr_id)
  REFERENCES Faculty (fid);
```

```
CREATE TABLE Faculty (
  fid INT,
  flast VARCHAR(50),
  ffirst VARCHAR(50),
  fmi CHAR(1),
  fdept CHAR(3),
  fsalary NUMERIC(10,2),
  fmgr_id INT
);
```

ALTER TABLE

```
CREATE TABLE Section (
  scall INT,
  sdept CHAR(3),
  scourse INT,
  sid INT
);
```

```
ALTER TABLE Section
  ADD PRIMARY KEY (scall);

ALTER TABLE Section
  MODIFY sdept CHAR(3) NOT NULL;

ALTER TABLE Section
  MODIFY scourse INT NOT NULL;

ALTER TABLE Section
  ADD CONSTRAINT fk_sec_fac
  FOREIGN KEY (sid)
  REFERENCES Faculty (fid);
```

```
ALTER TABLE Section
  ADD CONSTRAINT fk_sec_crs
  FOREIGN KEY (sdept,scourse)
  REFERENCES Course (cdept,ccourse);
```

Example 2:

```
DROP TABLE IF EXISTS DeptBgtTemp;
CREATE TEMPORARY TABLE IF NOT EXISTS DeptBgtTemp
  LIKE Department;
```

```
mysql> DESCRIBE DeptBgtTemp;
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| ddept          | varchar(3)    | NO   | PRI | NULL    |       |
| dname          | varchar(100)  | NO   |     | NULL    |       |
| dsalary_budget | decimal(15,2) | YES  |     | NULL    |       |
+----------------+---------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM DeptBgtTemp;
Empty set (0.00 sec)
```

Example 2:

```
ALTER TABLE DeptBgtTemp
  DROP COLUMN dname,
  ADD COLUMN (over_bgt TINYINT(1) NOT NULL,
              fid INT,
              fsalary NUMERIC(10,2),
              fsalary_sum NUMERIC(10,2)),
  DROP PRIMARY KEY,
  ADD PRIMARY KEY (ddept, fid),
  ADD INDEX over_bgt_idx (over_bgt);
```

```
mysql> DESCRIBE DeptBgtTemp;
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| ddept          | varchar(3)    | NO   | PRI | NULL    |       |
| dsalary_budget | decimal(15,2) | YES  |     | NULL    |       |
| over_bgt       | tinyint(1)    | NO   | MUL | NULL    |       |
| fid            | int(11)       | NO   | PRI | NULL    |       |
| fsalary        | decimal(10,2) | YES  |     | NULL    |       |
| fsalary_sum    | decimal(10,2) | YES  |     | NULL    |       |
+----------------+---------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

Example 1:

```
INSERT INTO DeptBgtTemp
  SELECT ddept,
         dsalary_budget,
         fsalary_sum,
         fsalary_sum > dsalary_budget AS over_bgt,
         fid,
         fsalary
  FROM (
    (SELECT dept, SUM(sal) AS fsalary_sum
     FROM (
       SELECT Department.ddept AS dept, 0 AS sal
         FROM Department
       UNION ALL
       SELECT fdept AS dept, SUM(fsalary) sal
         FROM Faculty
         GROUP BY fdept) AS t1
       GROUP BY dept) AS SalSums
    INNER JOIN Department
      ON (SalSums.dept = ddept)
    INNER JOIN Faculty
      ON (ddept = fdept)
  );
```

Example 1:

```
INSERT INTO DeptBgtTemp
  SELECT ddept,
         dsalary_budget,
         fsalary_sum,
         fsalary_sum > dsalary_budget AS over_bgt,
         fid,
         fsalary
    FROM (
      (SELECT dept, SUM(sal) AS fsalary_sum
       FROM (
         SELECT Department.ddept AS dept, 0 AS sal
           FROM Department
         UNION ALL
         SELECT fdept AS dept, SUM(fsalary) sal
           FROM Faculty
           GROUP BY fdept) AS t1
         GROUP BY dept) AS SalSums
     INNER JOIN Department
       ON (SalSums.dept = ddept)
     INNER JOIN Faculty
       ON (ddept = fdept)
  );
```

Execution plan for the SELECT statement:

```
+----+-------------+------------+------------+--------+---------------+-------------+---------+-------------------------------+------+----------+-------------------------------+
| id | select_type | table      | partitions | type   | possible_keys | key         | key_len | ref                           | rows | filtered | Extra                         |
+----+-------------+------------+------------+--------+---------------+-------------+---------+-------------------------------+------+----------+-------------------------------+
|  1 | PRIMARY     | Faculty    | NULL       | ALL    | fk_fac_dpt    | NULL        | NULL    | NULL                          |    8 |   100.00 | NULL                          |
|  1 | PRIMARY     | Department | NULL       | eq_ref | PRIMARY       | PRIMARY     | 5       | uncle_ted_sample.Faculty.fdept |    1 |   100.00 | NULL                          |
|  1 | PRIMARY     | <derived2> | NULL       | ref    | <auto_key0>   | <auto_key0> | 5       | uncle_ted_sample.Faculty.fdept |    2 |   100.00 | NULL                          |
|  2 | DERIVED     | <derived3> | NULL       | ALL    | NULL          | NULL        | NULL    | NULL                          |   14 |   100.00 | Using temporary; Using filesort |
|  3 | DERIVED     | Department | NULL       | index  | NULL          | PRIMARY     | 5       | NULL                          |    6 |   100.00 | Using index                   |
|  4 | UNION       | Faculty    | NULL       | index  | fk_fac_dpt    | fk_fac_dpt  | 5       | NULL                          |    8 |   100.00 | NULL                          |
+----+-------------+------------+------------+--------+---------------+-------------+---------+-------------------------------+------+----------+-------------------------------+
6 rows in set, 1 warning (0.00 sec)
```

Using the temp table, we only need to do this work once.

Example 1:

```
mysql> SELECT * FROM DeptBgtTemp;
+-------+----------------+----------+-------+----------+-------------+
| ddept | dsalary_budget | over_bgt | fid   | fsalary  | fsalary_sum |
+-------+----------------+----------+-------+----------+-------------+
| ADM   |      200000.00 |        0 | 52110 | 82000.00 |    82000.00 |
| BIO   |       40000.00 |        1 | 21004 | 21800.00 |    43200.00 |
| BIO   |       40000.00 |        1 | 31890 | 21400.00 |    43200.00 |
| CSI   |      180000.00 |        0 | 12058 | 48000.00 |   174900.00 |
| CSI   |      180000.00 |        0 | 22321 | 63400.00 |   174900.00 |
| CSI   |      180000.00 |        0 | 32000 | 63500.00 |   174900.00 |
| ENG   |       80000.00 |        1 | 12206 | 48000.00 |    81700.00 |
| ENG   |       80000.00 |        1 | 47862 | 33700.00 |    81700.00 |
+-------+----------------+----------+-------+----------+-------------+
8 rows in set (0.00 sec)
```

Example 2:

```
DROP TABLE IF EXISTS boxers;

CREATE TABLE boxers (
  rank INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100))
ENGINE=InnoDB
PARTITION BY RANGE (rank) PARTITIONS 3 (
PARTITION P1 VALUES LESS THAN (2)
  DATA DIRECTORY = '/demo_data/boxers_1/',
PARTITION P2 VALUES LESS THAN (4)
  DATA DIRECTORY = '/demo_data/boxers_2/',
PARTITION P3 VALUES LESS THAN MAXVALUE
  DATA DIRECTORY = '/demo_data/boxers_3/');

INSERT INTO boxers (name)
VALUES ('Ray Robinson'),
       ('Muhammad Ali'),
       ('Henry Armstrong'),
       ('Joe Louis'),
       ('Willie Pep'),
       ('Roberto Duran'),
       ('Benny Leonard'),
       ('Jack Johnson'),
       ('Jack Dempsey'),
       ('Sam Langford');
```

Example 2:

```
root@3f2d19876569:/# ls -lat /var/lib/mysql/sandbox
total 36
drwxr-xr-x 39 mysql mysql 4096 Oct 17 11:37 ..
drwxr-x---  2 mysql mysql 4096 Oct 17 05:03 .
-rw-r-----  1 mysql mysql   43 Oct 17 05:03 boxers#P#P3.isl
-rw-r-----  1 mysql mysql   43 Oct 17 05:03 boxers#P#P2.isl
-rw-r-----  1 mysql mysql   43 Oct 17 05:03 boxers#P#P1.isl
-rw-r-----  1 mysql mysql 8590 Oct 17 05:03 boxers.frm
-rw-r-----  1 mysql mysql   65 Oct  6 12:45 db.opt

root@8014faebf403:/# tree /demo_data/
/demo_data/
|-- boxers_1
|   -- sandbox
|       -- boxers#P#P1.ibd
|-- boxers_2
|   -- sandbox
|       -- boxers#P#P2.ibd
`-- boxers_3
    -- sandbox
        -- boxers#P#P3.ibd

           6 directories, 3 files

root@3f2d19876569:/# cat /var/lib/mysql/sandbox/boxers#P#P1.isl
/demo_data/boxers_1/sandbox/boxers#P#P1.ibd
```

Example 2:

```
root@3f2d19876569:/# ls -lat /var/lib/mysql/sandbox
total 36
drwxr-xr-x 39 mysql mysql 4096 Oct 17 11:37 ..
drwxr-x---  2 mysql mysql 4096 Oct 17 05:03 .
-rw-r-----  1 mysql mysql   43 Oct 17 05:03 boxers#P#P3.isl
-rw-r-----  1 mysql mysql   43 Oct 17 05:03 boxers#P#P2.isl
-rw-r-----  1 mysql mysql   43 Oct 17 05:03 boxers#P#P1.isl
-rw-r-----  1 mysql mysql 8590 Oct 17 05:03 boxers.frm
-rw-r-----  1 mysql mysql   65 Oct  6 12:45 db.opt

root@3f2d19876569:/# ls -lat /var/lib/mysql/uncle_ted_sample/
total 516
drwxr-xr-x 39 mysql mysql   4096 Oct 17 11:37 ..
-rw-r-----  1 mysql mysql 131072 Sep 26 15:59 Faculty.ibd
-rw-r-----  1 mysql mysql 131072 Sep 26 15:59 Section.ibd
-rw-r-----  1 mysql mysql  98304 Sep 26 15:59 Course.ibd
-rw-r-----  1 mysql mysql  98304 Sep 26 15:59 Department.ibd
drwxr-x---  2 mysql mysql   4096 Sep 26 15:59 .
-rw-r-----  1 mysql mysql   8658 Sep 26 15:59 Section.frm
-rw-r-----  1 mysql mysql   8756 Sep 26 15:59 Faculty.frm
-rw-r-----  1 mysql mysql   8630 Sep 26 15:59 Course.frm
-rw-r-----  1 mysql mysql   8644 Sep 26 15:59 Department.frm
-rw-r-----  1 mysql mysql     65 Sep 26 15:59 db.opt
```

DROP TABLE

```
DROP [TEMPORARY] TABLE [IF EXISTS]
    tbl_name [, tbl_name] ...
    [RESTRICT | CASCADE]
```

DROP TABLE

`DROP TABLE` removes one or more tables.

You must have the `DROP` privilege for each table.

All table data and the table definition are *removed*, ***so be careful with this statement!***

If any of the tables named in the argument list do not exist, MySQL returns an error indicating by name which non-existing tables it was unable to drop, but it also drops all of the tables in the list that do exist.

DROP TABLE

`DROP TABLE` removes one or more tables.

You must have the `DROP` privilege for each table.

All table data and the table definition are *removed*, **so *be careful* with this statement!**

If any of the tables named in the argument list do not exist, MySQL returns an error indicating by name which non-existing tables it was unable to drop, but it also drops all of the tables in the list that do exist.

Use `IF EXISTS` to prevent an error from occurring for tables that do not exist. A `NOTE` is generated for each nonexistent table when using `IF EXISTS`.

DROP TABLE

`RESTRICT` and `CASCADE` are permitted to make porting easier. In MySQL 5.7, they do nothing.

`DROP TABLE` automatically commits the current active transaction, unless you use the `TEMPORARY` keyword.

The `TEMPORARY` keyword has the following effects:

- The statement drops only `TEMPORARY` tables.
- The statement does not end an ongoing transaction.
- No access rights are checked. (A `TEMPORARY` table is visible only to the session that created it, so no check is necessary.)

Using `TEMPORARY` is a good way to ensure that you do not accidentally drop a non-`TEMPORARY` table.

To be continued...

Part two of the SQL Data Definition slides will include DML for Views, Indices, and Users.