MySQL supports several different storage engines. When you create a table you can specify the storage engine to be used or accept the default. I am assuming in class that we will be using the InnoDb storage engine.

The storage engine affects the way that the data is stored in the tables and whether or not the table supports a variety of features.

This is an example for the differences between the storage engines. We have discussed referential integrity in this unit. I am creating a pair of tables with a parent table and a child table using the MyISAM engine and another pair using the InnoDb engine.

Demo 01: The InnoDb engine

```
Create table z_db_parent (dept int primary key ) engine InnoDb;

Create table z_db_child (
  emp  int primary key
, dept int
, constraint fk_2 foreign key (dept) references z_db_parent (dept) )
engine InnoDb;

Insert into z_db_parent (dept )     values ( 200);
Query OK, 1 row affected (0.02 sec)

Insert into z_db_child (emp, dept) values (20, 200);
Query OK, 1 row affected (0.02 sec)

Insert into z_db_child (emp, dept) values (22, 22);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails
(`a_testbed`.`z_db_child`, CONSTRAINT `fk_2` FOREIGN KEY (`dept`) REFERENCES `z_db_parent`
(`dept`))

select * from z_db_child;
+-----+------+
| emp | dept |
+-----+------+
|  20 |  200 |
+-----+------+
1 row in set (0.00 sec)
```

The tables are created; I entered a row in the parent table. Then I entered one row in the child table with a valid foreign key and that was accepted. The last insert was rejected since I attempted to insert a row with a dept of 22 which does not match a parent row. We want that insert to fail since we do not want a row in the child table that claims a parent row that does not exist.

Demo 02: The MyISAM engine

```
Create table z_sam_parent  ( dept  int primary key ) engine MyISAM;
Create table z_sam_child (emp int primary key, dept int
, constraint fk_1 foreign key (dept) references z_sam_parent  (dept) )
engine MyISAM;

Insert into z_sam_parent  (dept  ) values ( 100);
Query OK, 1 row affected (0.02 sec)

Insert into z_sam_child (emp, dept) values (10, 100);
Query OK, 1 row affected (0.02 sec)
```

```
Insert into z_sam_child (emp, dept) values (11, 11);
Query OK, 1 row affected (0.02 sec)

select * from z_sam_child;
+-----+------+
| emp | dept |
+-----+------+
|  10 |  100 |
|  11 |   11 |
+-----+------+
2 rows in set (0.00 sec)
```

This is a problem. I created tables with the same sql except for specifying the MyISAM engine. With the MyISAM table both of the child rows were accepted even though the last insert did not follow referential integrity.

If you use the SHOW CREATE TABLE command for each table you can see that InnoDb table kept the foreign key constraint while the MyISAM table dropped that constraint. The MyISAM engine does not support foreign keys and the create table statement did not tell you that. (It would be nice to get a warning that the dbms is throwing away part of your code!)

Demo 03:

```
show create table z_sam_child \G
*************************** 1. row ***************************
       Table: z_sam_child
Create Table: CREATE TABLE `z_sam_child` (
  `emp` int(11) NOT NULL,
  `dept` int(11) DEFAULT NULL,
  PRIMARY KEY (`emp`),
  KEY `fk_1` (`dept`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1



show create table z_db_child \G
*************************** 1. row ***************************
       Table: z_db_child
Create Table: CREATE TABLE `z_db_child` (
  `emp` int(11) NOT NULL,
  `dept` int(11) DEFAULT NULL,
  PRIMARY KEY (`emp`),
  KEY `fk_2` (`dept`),
  CONSTRAINT `fk_2` FOREIGN KEY (`dept`) REFERENCES `z_db_parent` (`dept`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Why would you want tables that do not enforce referential integrity? MyISAM tables have been faster than InnoDb (for one thing the system does not have to validate foreign keys!) The thinking in MySQL was that the application level of the system would handle the validation.

Why would you want tables that do enforce referential integrity? To help keep your data clean.

It is also important to realize that there was no warning about the foreign key constraint with the MyISAM table. This is a MySQL feature. The constraint was silently dropped. That could easily be a problem if you are not aware of this feature. It can also be a problem if your installation defaults to MyISAM tables.

The InnoDb tables are closer to standard SQL than the MyISAM tables which is why they are required for this class.

The SQL that I give you for the major tables includes the engine=InnoDb phrase. When I include Create tables statements in the notes, I do not always include that phrase- but if InnoDb is not your default engine you may need to add that phrase.

One consequence of having different storage engines is that when you go to various web sites the author might not specify which engine they are using. The MyISAM engine was the default in earlier versions of MySQL and many web page authors are using that engine. The InnoDB engine is the default with MySQL version 5.5.