

CS122 Using Relational Databases and SQL DDL and DML

Chengyu Sun
California State University, Los Angeles

Queries vs. Updates

- ◆ Queries – statements that do not change the data
- ◆ *Updates*
 - Create, delete, and change tables
 - Create, delete, and change data in the tables

SQL

- ◆ Data Definition Language (DDL)
 - CREATE, DROP, ALTER
- ◆ Data Manipulation Language (DML)
 - SELECT, INSERT, DELETE, UPDATE
- ◆ Data Control Language (DCL)
 - GRANT, REVOKE
 - COMMIT, ROLLBACK, SAVEPOINT

Create a Table

```
create table table_name (  
    column_name column_type,  
    column_name column_type,  
    ...  
    column_name column_type  
);
```

Create the Products Table

```
create table products (  
    id integer,  
    category char(3),  
    description varchar(4096),  
    price decimal(10,2)  
);
```

Delete a Table

```
drop table table_name;
```

MySQL only:

```
drop table if exists table_name;
```

Naming Conventions

- ◆ Use plural form for table names
- ◆ Use singular form for column names
- ◆ Use underscore to concatenate multiple words, e.g. `employee_id`
 - Do not use mixed cases in names (e.g. `ArtistName`) because many DBMS treat names as case-insensitive

Data Type

- ◆ Determines the storage required for a field
- ◆ Common data types
 - String types
 - Numeric types
 - Date and time types
 - Other types

String Types

- ◆ `char (n)`
 - Fixed-length strings
 - Max length `n`
- ◆ `varchar (n)`
 - Variable-length strings
 - Max length `n`
- ◆ `text`
 - For articles, essays, ...

char(6)

S	U	N			
C	H	E	N	G	Y

varchar(6)

S	U	N			
C	H	E	N	G	Y

Numeric Types

- ◆ Integer types
 - `integer, int`
 - Variations: `smallint, bigint, long, ...`
- ◆ Floating-point types
 - `real`
 - Variations: `float, double, ...`
- ◆ Arbitrary precision number
 - `decimal (m, n)`
 - `numeric (m, n)`
- ◆ Boolean
 - `boolean, bool`

Date and Time Types

- ◆ `date` – YYYY-MM-DD
- ◆ `time` – HH:MM:SS
- ◆ `datetime` – YYYY-MM-DD HH:MM:SS
- ◆ `timestamp` – YYYY-MM-DD HH:MM:SS

MySQL Storage Engines

- ◆ MyISAM
 - *Default*
 - Does *not* support transactions and some integrity constraints
- ◆ InnoDB
 - Supports transactions and integrity constraints
- ◆ Memory, BDB, NDB, ...

```
create table products (
  id integer,
  category char(3),
  description varchar(4096),
  price decimal(10,2)
) Engine=InnoDB;
```

Data Integrity Constraints

- ◆ Not NULL
- ◆ Default
- ◆ Unique
- ◆ Primary key
 - Unique + Not NULL
 - Only one primary key per table
- ◆ Check

Column Constraint Syntax

```
create table products (  
    id            integer primary key,  
    category     char(3) not null,  
    description   varchar(4096) default 'Some product',  
    price        decimal(10,2) not null check(price > 0)  
);
```

Table Constraint Syntax

```
create table products (  
    id            integer,  
    category     char(3) not null,  
    description   varchar(4096) default 'Some product',  
    price        decimal(10,2) not null,  
    primary key (id),  
    check (price > 0)  
);
```

Named Constraints

```
create table products (  
    id            integer,  
    category     char(3) not null,  
    description   varchar(4096) default 'Some product',  
    price        decimal(10,2) not null,  
    constraint products_pk primary key (id),  
    constraint products_price_gt0 check (price > 0)  
);
```

Foreign Key Constraints

orders

id	customer_id	date_ordered	date_shipped
----	-------------	--------------	--------------

order_details

order_id	product_id	quantity
----------	------------	----------

- ◆ Ensure that the value of `order_id` is valid, i.e. the value appears in the `id` column of the `orders` table

Foreign Key Constraint Example

```
create table order_details (  
    order_id     integer not null references orders(id),  
    product_id   integer not null,  
    quantity     integer not null check(quantity>0),  
    foreign key (product_id) references products(id),  
    primary key (order_id, product_id)  
);
```

Modify a Table

```
alter table table_name operation;
```

◆ Common operations

- Add, remove, rename, retype columns
- Add, remove constraints

◆ *Exactly what operation are supported depends on the DBMS*

- <http://dev.mysql.com/doc/refman/5.5/en/alter-table.html>

Alter Table Example

- ◆ Split the `address` column in the `customers` table into four columns:
`street, city, state, zip`

Populate Tables with Data

```
insert into table values (value1, value2, ...);
```

```
insert into table (field, ...) values (value, ...);
```

◆ Example: insert the following data into the `Products` table

- WD 500G Hard drive for \$100.00
- Nvidia 7600GS video card for \$104.99

Insert the Results of a Query

```
insert into table select_query;
```

```
insert into table (field, ...) select_query;
```

Delete Data

```
delete from table [where condition(s)];
```

◆ Examples:

- Delete the product with `id=2`
- Delete all CPU products
- Delete all products

Update Data

```
update table set field=value [, ...] where condition(s);
```

◆ Examples:

- Change the price of Intel Core 2 Duo to \$149.99
- Change the last name of Jane from DOE to Doe
- Raise the price of all CPU products by 10%

Summary

- ◆ Remember what can be done
 - Create, alter, drop tables
 - ◆ Data types
 - ◆ Data integrity constraints
 - Insert, update, delete data
- ◆ Look up DBMS manual for the syntax