SQL Data Manipulation (DML)
SQL Data Definition (DDL)

Today's Topics

- How did it go with the SQL exercises – any question, tips or tricks?

- We will work with
  - SQL DML statements
    - INSERT, UPDATE and DELETE

  - SQL DDL statements
    - CREATE TABLE
    - CREATE VIEW (mostly in study point assignment ☺)

  - Logical backup of tables

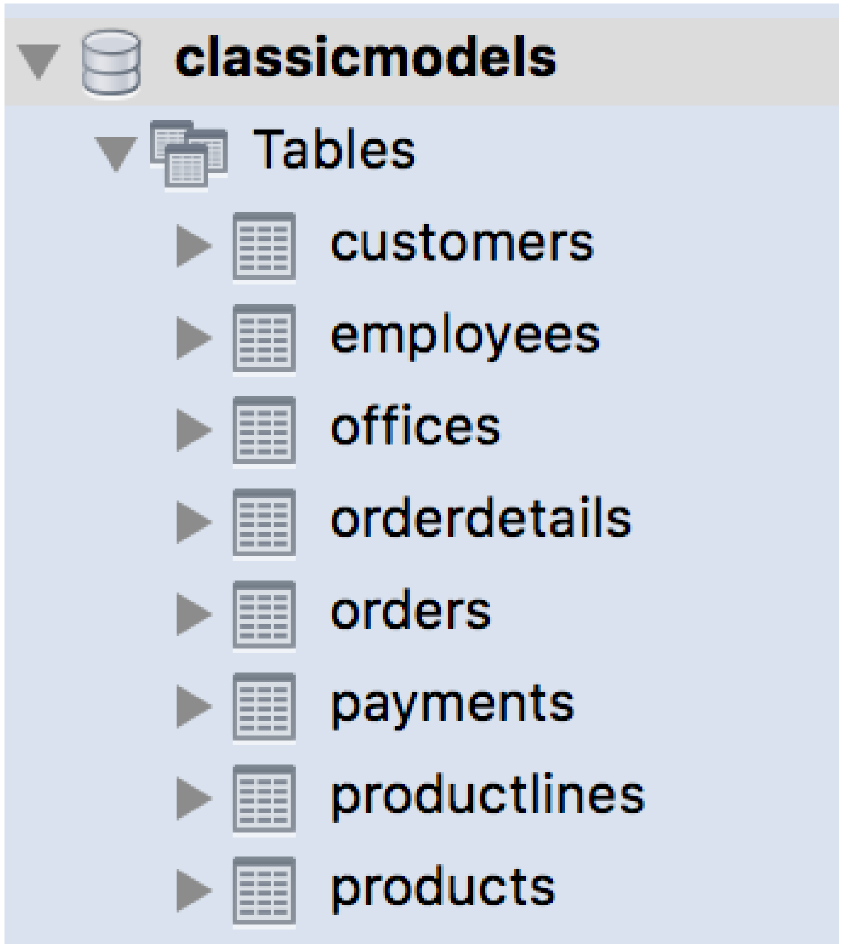
Resources

There are different resources about SQL and MySQL online. Three good ones are:

- http://www.w3schools.com/sql/default.asp
- http://www.mysqltutorial.org/basic-mysql-tutorial.aspx
- http://www.mysqltutorial.org/basic-mysql-tutorial.aspx

Wikipedia also has good examples for most SQL commands

Database & Tables


classicmodels
Tables
customers
employees
offices
orderdetails
orders
payments
productlines
products

A database describes the tables of a database. It has a name (here "classicmodels")

A database can have more than one table

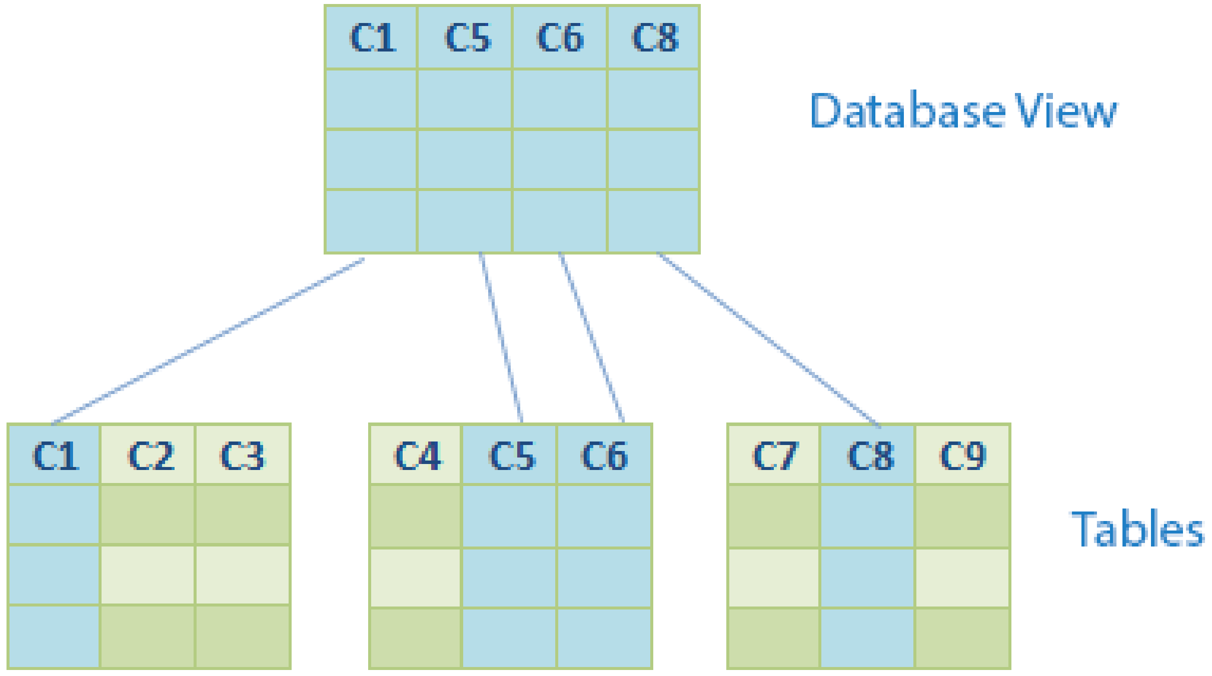
Each table has a unique name

What other object types can a database have?

Nice, but not need to know ☺

Another database object type: View

- A view is a virtual table computed or collated dynamically from data in the database when access to that view is requested.
- Example:



Nice, but not need to know ☺

Another database object type: Stored procedures

- Stored procedure
  - Logic taking place on database server instead of in the application
- It is a matter of
  - database-oriented software developers and in-memory application software developers
  - performance versus maintenance

cphbusiness

Nice, but not need to know ☺

Pros & Cons on Stored procedures

Pro

- SQL has extremely powerful capabilities for querying the database (i.e. better performance)
- SQL and application language are different programming skills (a problem ?)

Con

- SQL queries often embed domain logic, which goes against the basic principles of a layered enterprise application architecture.
- Database portability
- Testability

You can read more about pros and cons here:

https://www.martinfowler.com/articles/dblogic.html

.NET example of how to call procedure: https://dev.mysql.com/doc/connector-net/en/connector-net-tutorials-stored-procedures.html

Example of how to define procedure: http://www.mysqltutorial.org/mysql-if-statement/

cphbusiness

Aggregate functions 1

- You were asked to find the highest profit amongst products (i.e. MSRP-buyPrice)

```
SELECT max(msrp-buyprice) FROM classicmodels.products
```

- What if you want to output product name instead of the profit?

```
SELECT productname ???
```

Aggregate functions 2

Group by examples

List the name of the customer with the **highest** credit limit **in each country**. Order the list alphabetically by country name.

```
SELECT customername FROM customers
WHERE  creditlimit in (SELECT max(creditlimit)
   FROM classicmodels.customers group by country)
group by country
order by country
```

List the **total** quantity in stock for **each product scale** that has a **total quantity above 1000**

```
SELECT productscale, sum(quantityInStock)
from products
group by productscale
having sum(quantityinstock) > 1000
```

Data Definition Language (DDL)

Create Table Example

```
create table EMP (
   EMPNO integer(4) not null,
   ENAME varchar(30) not null,
   JOB varchar(10),
   MGR integer (4),
   HIREDATE date,
   SAL decimal(7,2),
   DEPTNO integer (2)
);
```

empno	ename	job	mgr	hiredate	sal	deptno
7369	SMITH	CLERK	7902	12/17/1980	800	20
7499	ALLEN	SALESMAN	7698	02/20/1981	1600	30
7521	WARD	SALESMAN	7698	02/22/1981	1250	30
7566	JONES	MANAGER	7839	04-02-1981	2975	20
7654	MARTIN	SALESMAN	7698	09/28/1981	1250	30
7698	BLAKE	MANAGER	7839	05-01-1981	2850	30
7782	CLARK	MANAGER	7839	06-09-1981	2450	10
7788	SCOTT	ANALYST	7566	04/19/1987	3000	20
7839	KING	PRESIDENT		11/17/1981	5000	10
7844	TURNER	SALESMAN	7698	09-08-1981	1500	30
7876	ADAMS	CLERK	7788	05/23/1987	1100	20
7900	JAMES	CLERK	7698	12-03-1981	950	30
7902	FORD	ANALYST	7566	12-03-1981	3000	20
7934	MILLER	CLERK	7782	01/23/1982	1300	10

Constraints – primary and foreign key

```
create table EMP (
   EMPNO integer(4) not null,
   ENAME varchar(30) not null,
   JOB varchar(10),
   MGR integer(4),
   HIREDATE date,
   SAL decimal(7,2),
   DEPTNO integer(2),
   PRIMARY KEY (empno),
   CONSTRAINT emp_fk FOREIGN KEY (deptno)
    REFERENCES dept(deptno)
)
```

empno	ename	job	mgr	hiredate	sal	deptno
7369	SMITH	CLERK	7902	12/17/1980	800	20
7499	ALLEN	SALESMAN	7698	02/20/1981	1600	30
7521	WARD	SALESMAN	7698	02/22/1981	1250	30
7566	JONES	MANAGER	7839	04-02-1981	2975	20
7654	MARTIN	SALESMAN	7698	09/28/1981	1250	30
7698	BLAKE	MANAGER	7839	05-01-1981	2850	30
7782	CLARK	MANAGER	7839	06-09-1981	2450	10
7788	SCOTT	ANALYST	7566	04/19/1987	3000	20
7839	KING	PRESIDENT		11/17/1981	5000	10
7844	TURNER	SALESMAN	7698	09-08-1981	1500	30
7876	ADAMS	CLERK	7788	05/23/1987	1100	20
7900	JAMES	CLERK	7698	12-03-1981	950	30
7902	FORD	ANALYST	7566	12-03-1981	3000	20
7934	MILLER	CLERK	7782	01/23/1982	1300	10

deptno	dname	loc
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

cphbusiness

Logical backup of data

SELECT INTO statement copies data from one table and inserts it into a new table.

See more here: https://www.w3schools.com/sql/sql_select_into.asp

Data Definition Language (DML) – insert, update, delete

Insert data (new rows)

insert into <table> [(<column i, . . . , column j>)]
values (<value i, . . . , value j>);

Example:

```
insert into PROJECT(PNO,PNAME,PERSONS,BUDGET, PSTART)
values(313, ’DBS’, 7411, 150000.42, ’10-OCT-16’);
```

Or

```
insert into PROJECT
values(313,’DBS’,7411,null,150000.42 ’10-OCT-16’, null);
```

Insert as a copy from another table

insert into <table> [(<column i, . . . , column j>)]
    <query>

Example:

```
insert into OLDEMP (ENO, HDATE)
select EMPNO, HIREDATE from EMP
where HIREDATE < '01-JAN-17';
```

Update data (change rows)

update <table>
set <column i> = <expression i>, . . . ,
<column j> = <expression j> [where <condition>];

Examples:

```
update EMP
set JOB = 'MANAGER', DEPTNO = 20, SAL = SAL +1000
where ENAME = 'JONES';

update EMP
Set SAL = SAL * 1.15
where DEPTNO in (10,30);
```

cphbusiness

Deletion

Delete data (rows):

delete **from** <table> [where <condition>];

```
delete from PROJECT
where BUDGET < 10000;
```

Delete table:

```
drop emp;
```

Commit and Rollback

- A sequence of database modifications, i.e., a sequence of insert, update, and delete statements, is called a transaction.

- Modifications of data in tables are <u>temporarily</u> stored in the database system.

- They become permanent only after the statement <u>commit</u>;

- To undo modifications, one has to use the statement <u>rollback</u>;
  - Only if you haven't committed already ☺

Exercises

- See document in exercise folder