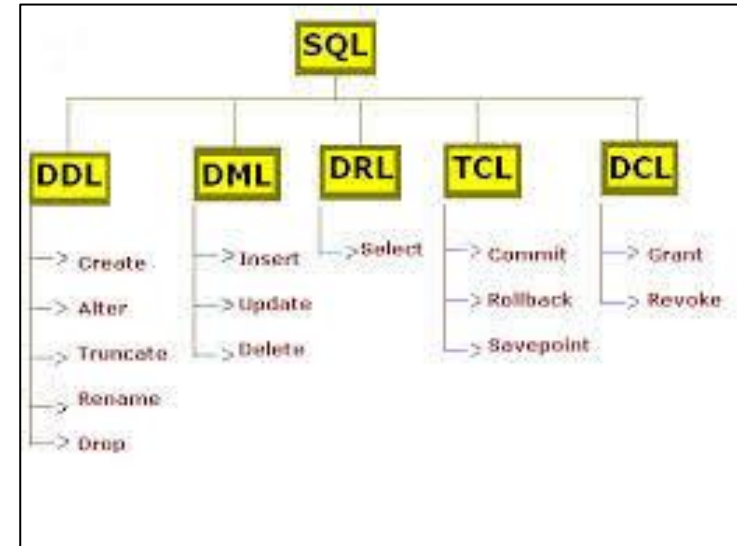


ICS 129 DATABASES

BY: Derek Payne

Structured Query Language

Structured Query Language(SQL) is a database language that is used to create and perform many different tasks on existing databases. The SQL commands that are used to carry out these tasks are Data Definition Language, Data Manipulation Language, and Data Query Language.



Data Definition Language

Data definition language(DDL) is SQL commands that are used to create database schema. DDL is also used to modify the structure of the data in the database. Create, drop, and alter are some of the main DDL commands that are used.

```
CREATE TABLE <table_name> (  
<Column_name_1>: <type> [PRIMARY KEY]  
    , <column_name_2>: <type>  
    );
```

Data Manipulation Language

Data Query Language

Data manipulation language(DML) consists of SQL commands that are used to manipulate data within the database. Insert, update, and delete are three examples of DML.

```
'INSERT INTO <name> ()
```

Data Query Language(DQL) is used to query data from the database. Select is the main command for DQL.

```
SELECT * FROM <name>
```

Data Types

VARCHAR(n)- Variable-length character which specifies the maximum length

CHAR(m)- Fixed length that has to be an exact amount of characters.

NUMERIC(t,d)- The t stands for the all of the digits, and the d is for how many of the digits are in a decimal place.

INT- Integer

Date- YYYY/MM/DD

Primary Key- Constraint that uniquely identifies each record in a table.

Foreign Key- A foreign key is used to link two tables together and it is a primary key in another table.

Custom Furniture Oahu

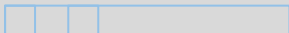
A small company that creates custom furniture just recently started up, and is looking to organize their job projects and employee's.

JOB_NUM	JOB_NAME	EQUIPMENT	EMP_NAME	EMP_SKILL	EMP_HOURS	EMP_PRICE
1	Wood Table	Saw, Drill, Paint,	Alex Smith	Carpentry	8.50	23.50
1	Wood Table	Deliver	James Trech	Hauler	2.00	16.50
2	Marble Table	Cutting, Measuring, Polishing	Jeremy Foss	Carpentry	10.00	28.50
2	Marble Table	Deliver	James Trech	Hauler	3.00	16.50
3	Granite Table	Cutting, Measuring, Polishing	Naomi Lin	Carpentry	8.50	30.50
3	Granite Table	Deliver	James Trech	Hauler	4.00	16.50

```
use p01;
CREATE TABLE COMPANY(
  JOB_NUM Varchar(30)
  , JOB_NAME VARCHAR(40)
  , EQUIPMENT VARCHAR(40)
  , EMP_NAME VARCHAR(30)
  , EMP_SKILL Varchar(40)
  , EMP_HOURS NUMERIC(5,2)
  , EMP_PRICE NUMERIC(5,2)
);

INSERT INTO COMPANY VALUES('1','Wood Table','Saw, Drill, Paint','Alex Smith','Carpentry','8.5','23.50');
INSERT INTO COMPANY VALUES('1','Wood Table','Deliver','James Trech','Hauler','2','16.50');
INSERT INTO COMPANY VALUES('2','Marble Table','Cutting, Measuring, Polishing','Jeremy Foss','Carpentry','10','28.50');
INSERT INTO COMPANY VALUES('2','Marble Table','Deliver','James Trech','Hauler','3','16.50');
INSERT INTO COMPANY VALUES('3','Granite Table','Cutting, Measuring, Polishing','Naomi Lin','Carpentry','8.5','30.5');
INSERT INTO COMPANY VALUES('3','Granite Table','Deliver','James Trech','Hauler','4','16.50');

select * from COMPANY;
```



Normalization

The tables on the right are showing the beginning of the normalization process.

```
Create table JOB (
```

```
    JOB_ID INT PRIMARY KEY
```

```
    , JOB_NAME VARCHAR(30)
```

```
);
```

```
INSERT INTO JOB VALUES('1', 'Wood Table');
```

```
INSERT INTO JOB VALUES('2', 'Marble Table');
```

```
INSERT INTO JOB VALUES('3', 'Granite Table');
```

```
Select * from JOB;
```

```
CREATE TABLE EQUIPMENT (
```

```
    EQUIP_ID CHAR(2) PRIMARY KEY
```

```
    , EQUIPMENT varchar(50)
```

```
);
```

```
INSERT INTO EQUIPMENT VALUES('A1', 'Saw, Drill, Paint,');
```

```
INSERT INTO EQUIPMENT VALUES('A2', 'Deliver');
```

```
INSERT INTO EQUIPMENT VALUES('A3', 'Cutting, Measuring, Polishing');
```

```
SELECT * FROM EQUIPMENT;
```

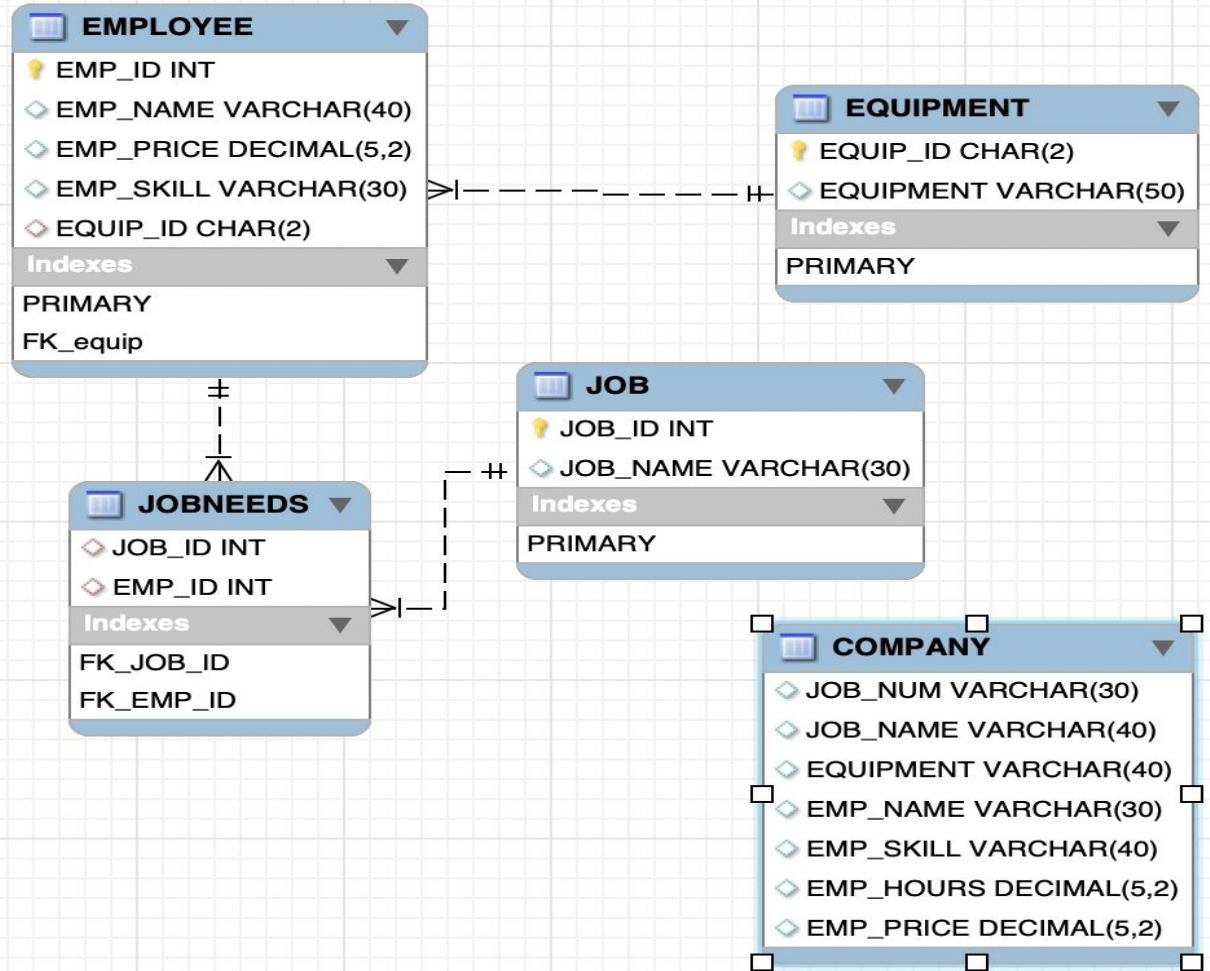
Normalization

Continuation of normalization

```
Create Table EMPLOYEE (  
    EMP_ID int PRIMARY KEY  
    , EMP_NAME VARCHAR(40)  
    , EMP_PRICE NUMERIC(5,2)  
    , EMP_SKILL VARCHAR(30)  
    , EQUIP_ID CHAR(2)  
);  
alter table EMPLOYEE add constraint FK_equip foreign key (EQUIP_ID) references EQUIPMENT (EQUIP_ID);  
  
INSERT INTO EMPLOYEE VALUES(1, 'Alex Smith', '23.50', 'Carpentry','A1');  
INSERT INTO EMPLOYEE VALUES(2, 'James Trech', '16.50', 'Hauler', 'A2');  
INSERT INTO EMPLOYEE VALUES(3, 'Jeremy Foss', '28.50', 'Carpentry', 'A3');  
INSERT INTO EMPLOYEE VALUES(4, 'Naomi Lin', '30.50', 'Carpentry', 'A3');  
  
SELECT * FROM EMPLOYEE;  
  
Create TABLE JOBNEEDS(  
    JOB_ID INT  
    , EMP_ID INT  
);  
-- add foreign keys user alter table  
alter table JOBNEEDS add constraint FK_JOB_ID foreign key (JOB_ID) references JOB (JOB_ID);  
alter table JOBNEEDS add constraint FK_EMP_ID foreign key (EMP_ID) references EMPLOYEE (EMP_ID);  
  
INSERT INTO JOBNEEDS VALUES(1,1);  
INSERT INTO JOBNEEDS VALUES(1,2);  
INSERT INTO JOBNEEDS VALUES(2,3);  
INSERT INTO JOBNEEDS VALUES(2,2);  
INSERT INTO JOBNEEDS VALUES(3,4);  
INSERT INTO JOBNEEDS VALUES(3,2);  
SELECT * FROM JOBNEEDS;
```

Entity Relationship Diagram(ERD)

This ERD is showing how the tables that were shown in the previous slide are related.



Resources

<https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/>

https://www.w3schools.com/sql/sql_foreignkey.asp

<https://beginnersbook.com/2015/05/normalization-in-dbms/>

<https://www.assignmentpoint.com/science/computer/features-structured-query-language.html>