

SQL Week 1

Goals of video 2 (Structured Query Language)

- define SQL,
- discuss how SQL differs from many other computer languages,
- explain the three primary ways SQL is used with the database, and
- compare and contrast the roles of a database administrator and a data scientist.
- Along with discussing the importance of knowing what SQL syntax you're using in a given database

What is SQL: This is the standard language for many relational database management systems and data manipulation. SQL is used often to query, insert, update, and modify data. At a basic level SQL is a method for communicating between you and the database.

What SQL is used for:

- It's used to **read and retrieve data**, so data is often stored in a database, and you want to retrieve it or read it. And you can use SQL as a means to be a translator for that.
- SQL is also used as a way to **write** data in a database. So if you need to write data in a table or insert new data, you can use SQL as a means to do this.
- And finally, it's used to **update** and insert new data.

A DBA is responsible for managing the entire database and guarding it.

A data scientist, on the other hand, is typically a user of that database. The DBA will be responsible for giving permissions to people and determining who has access to what data. They're often times responsible for managing the tables and creating them

Data Scientist uses SQL for: Retrieve data, may create their own table or test environment, combine multiple sources, write complex queries analysis.

Goals of video 3

- Explain the importance of understanding how the data in a database relates to one another, and describe what a database is at its core.
- Describe what a database is

***** Learn Data Modeling and ER diagrams! *****

A database is really a container that is usually a file or set of files and is used to organize and store all of the data. If you think of this in real world terms, It'd be like a filing system that has many cabinets along a wall. Within that system, within a database, we have tables, these tables are structured lists of data elements or specific data type.

Column: A single field in a table

Row: A record in a table

Goals of video 4

- Describe data Modeling
- Define relational database system
- Discuss advent of relational database in SQL

Data modeling is what we use to organize information for multiple tables and how they relate to each other together. This helps tremendously in providing structure to the information in the system. Usually a data model represents a **business process** and it can also help you understand a business process. And show **relationship between** business processes.

Data model: model for prediction built by data scientist.

Data model as a data table represented and organized in a database

There's been hierarchical, network, relational, entity, relational somatic, and NoSql.

Goals of video 5

- Define and describe both relational and transactional database models,
- Define entities, attributes and relationship,
- Describe and explain the difference between one-to-one, one-to-many and many-to-many relationships,
- Describe the use of a primary key in a database and
- Explain how ER diagram is used to document and illustrate relationships

A relational model is a database design that shows the relationships between the different tables and this is really used to optimize querying data, making it easy and intuitive to access the data.

Transactional, on the other hand, you can think of as a more operational database.

The building blocks for this relational model are really three simple things. We have entities which are a person, place, thing or event. These are very distinguishable. They are unique. They are distinct. For example, I could be an entity, Sadie St. Lawrence, and then we have attributes, which are characteristics of this entity. As an entity, it would be myself and then an attribute about me would be that I am female.

Then, the third building block of the model is the relationship. This describes the association among different entities. One-to-many, many-to-many, one-to-one are covered here.

One-to-many: customer to invoices

Many-to-many: student to classes

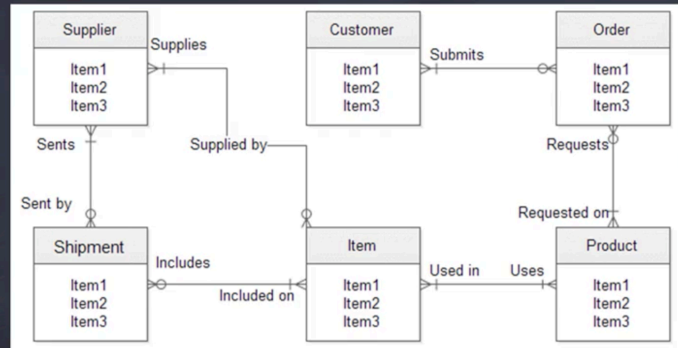
One-to-one: manager to store

An ER model then is composed of the entity types and the specific relationships that can exist between these entities

ER Diagrams

ER model

Is composed of entity types and specifies relationships that can exist between instances of those entity types



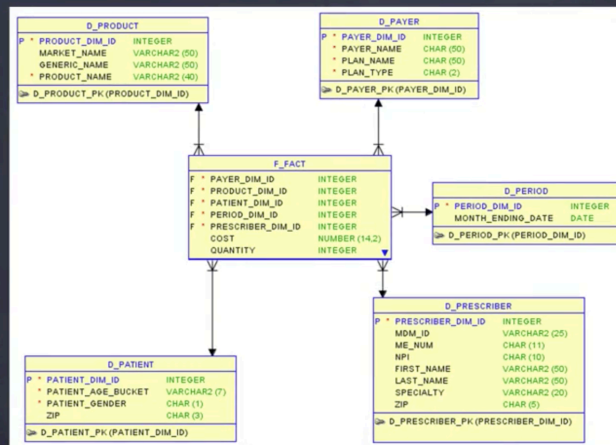
ER Diagrams

Show relationships

Business process

Represented visually

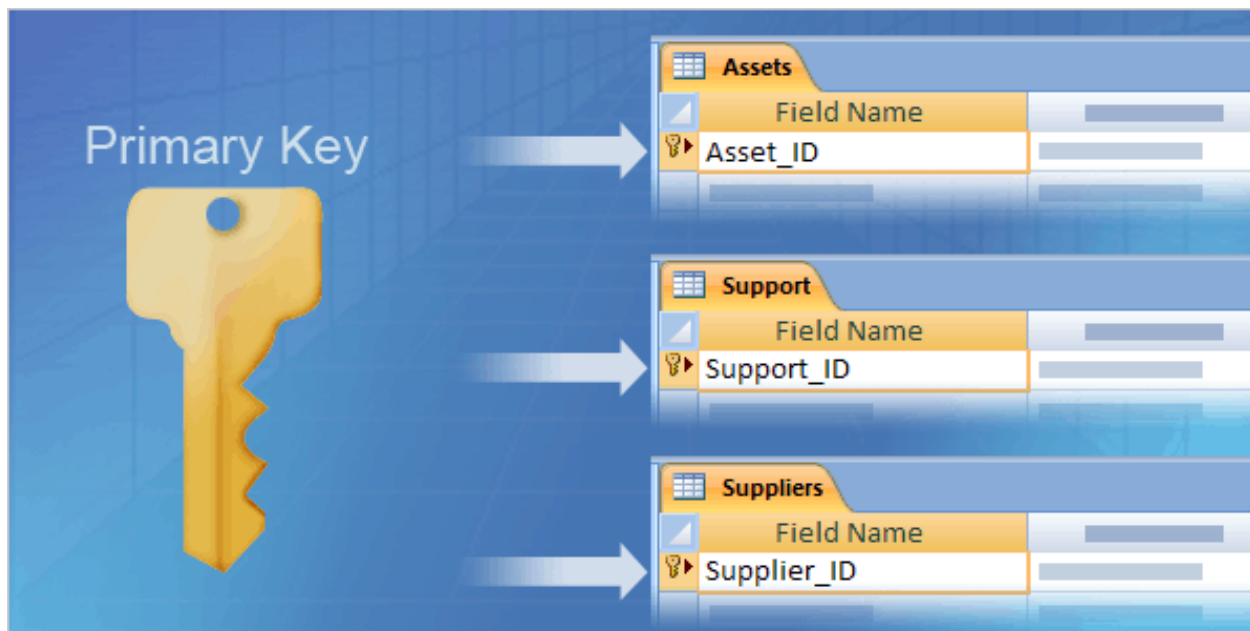
Show links (primary keys)



The primary key is a column or set of columns whose values uniquely identify every row in a table. Then this allows us to take those unique identities and then join it to another table.

Primary keys must contain UNIQUE values, and cannot contain NULL values.

A table can have only one primary key, which may consist of single or multiple fields.



Customer ID	Forename	Surname
1	Simon	Jones
2	Emma	Price
3	Laura	Jones
4	Jonathan	Hale
5	Emma	Smith

Simple primary key

Foreign keys are similar but in regard that one or more columns can be used together to identify a single row in the table.

The following source: [here](#)

- A FOREIGN KEY is a key used to link two tables together.
- A FOREIGN KEY is a field (or collection of fields) in one table that refers to the PRIMARY KEY in another table.
- The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.

"Persons" table:

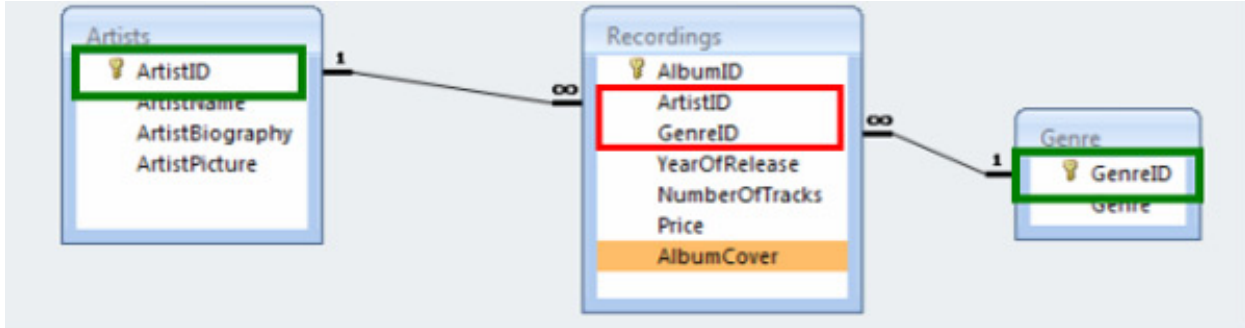
PersonID	LastName	FirstName	Age
1	Watkins	Ben	10
2	Johnson	David	12
3	Kevinson	Kevin	11
4	Whatever	Barack	9

"Orders" table:

OrderID	OrderNo	PersonId	OrderDate
1	1234	14	1/1/2011
2	45677	15	
3	123456	15	
4	43210	10	

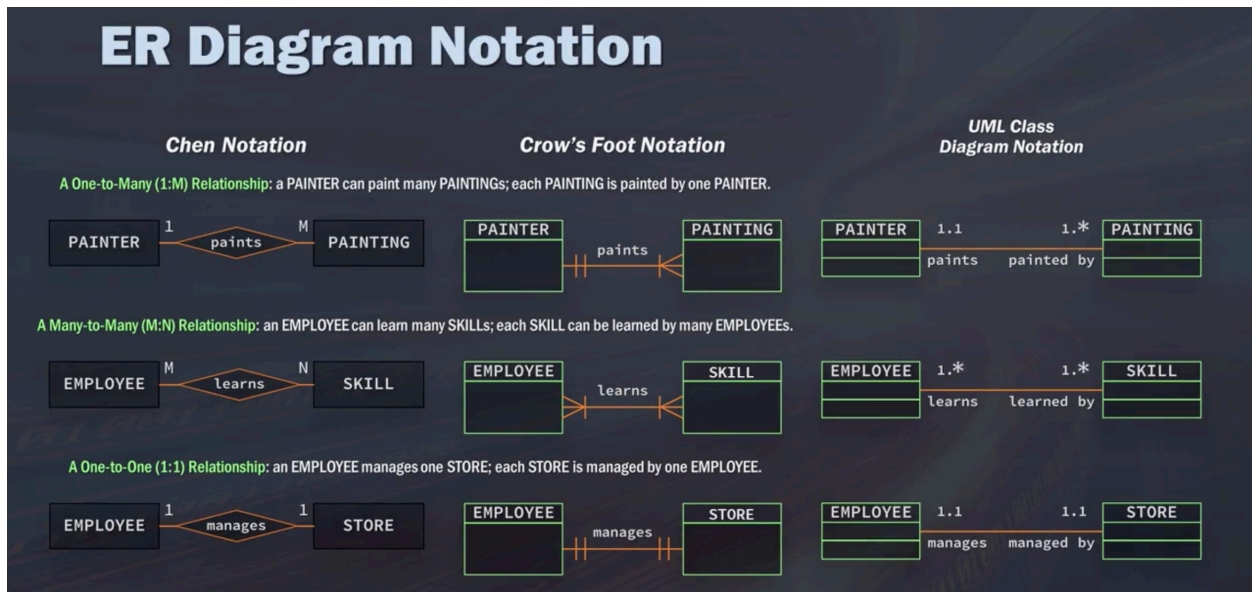
- Notice that the "PersonID" column in the "Orders" table points to the "PersonID" column in the "Persons" table.
- The "PersonID" column in the "Persons" table is the PRIMARY KEY in the "Persons" table.
- The "PersonID" column in the "Orders" table is a FOREIGN KEY in the "Orders" table.
- The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables.

- The FOREIGN KEY constraint also prevents invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the table it points to.



= primary key
 = foreign key

Relationship Notations:



SELECT statement

Goals:

- Write a basic SELECT statement
- Tell a database which table your data will come FROM
- SELECT either all or particular columns from a table in a query
- Limit the amount of data which is returned in a query.

```
SELECT column(s)_name  
FROM table_name
```

e.g. output:

```
SELECT prod_name      prod_name  
FROM products        Shampoo  
                     Cologne  
                     Camera
```

```
SELECT prod_name,  
       prod_id  
FROM products
```

```
SELECT *.  <--- The * wildcard selects all columns.  
FROM products
```

Limit the number of items we retrieve:

SQLite:

```
SELECT column(s)_name  
FROM table_name  
LIMIT number_of_records
```

ORACLE: WHERE ROWNUM<=5

Creating Tables:

Goals:

- discuss situations where it's beneficial to create new tables,
- create new tables within an existing database,
- write data in a new table, and
- define whether columns can accept **null** values or not

CREATE TABLE shoes

```
(  
Id.          char(10)          PRIMARY KEY,  
Brand        char(10)          NOT NULL,  
color        char(15)          NOT NULL,  
Price        decimal(8, 2)    NOT NULL,  
Desc         Varchar(750)     NULL  
)
```

- If in the last column we do not specify whether null is acceptable or not, it assumes, by default, the null type are accepted. If you choose NOT NULL for a column, then you will get error if you do not enter a value in that column when you are inserting data into it.
- PRIMARY KEY cannot be NULL.
- Null values are absence of anything. EMPTY STRINGS has actually a value in them.

INSERT INTO shoes

```
VALUES ( `1234`,  
        `Gucci`,  
        `Pink`,  
        NULL  
        )
```

OR

INSERT INTO shoes

```
(Id.,  
Brand,  
Color,  
Price,  
Desc  
)  
VALUES  
(  
`1234`, ... )
```

Creating TEMPORARY tables

Goals:

- Create temporary tables
- Describe limitations of temporary tables
- Discuss strategies for researching syntax for particular database management systems

Temp. Tables will be deleted when current session terminates.

```
CREATE TEMPORARY TABLE Sandals AS
(
  SELECT *
  FROM Shoes
  WHERE shoe_type = `sandals`
)
```

Adding Comments to SQL

```
SELECT shoe_id
      --, brand_id (comment out one line)
      , shoe_name
FROM shoes
```

- comment out several lines

```
SELECT shoe_id
      /*, brand_id (comment out one line)
      , shoe_name
      */
FROM shoes
```