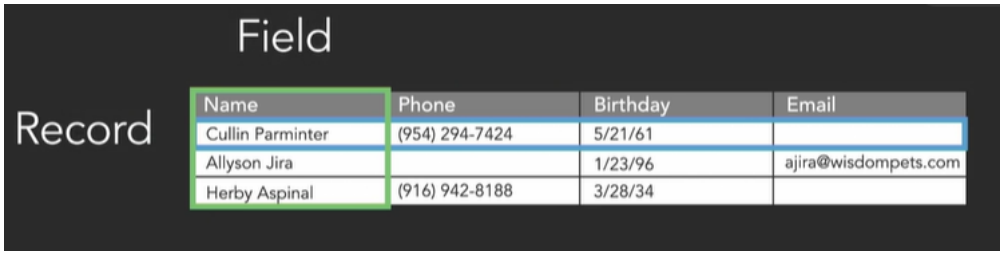
- 07 - Programming Foundations: Databases

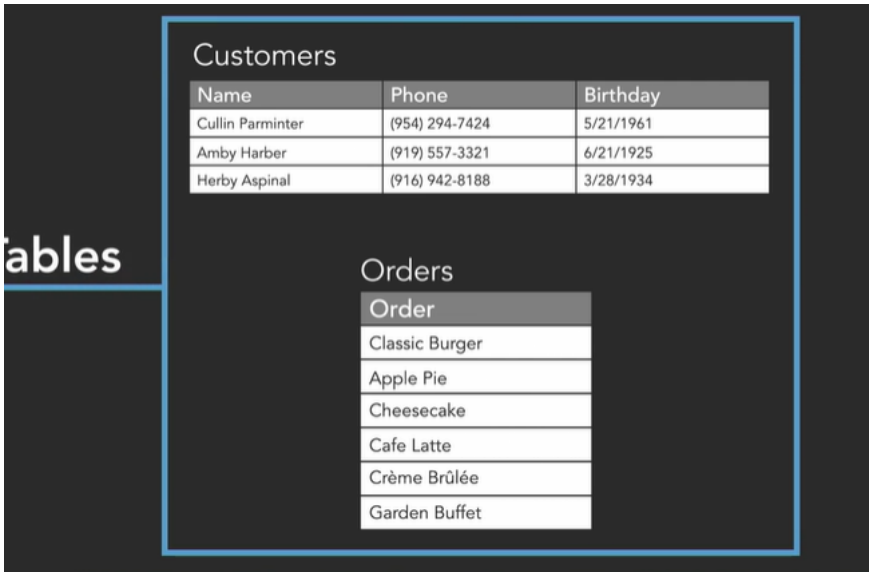
Certificate	CertificateOfCompletion_Programming Foundations Databases.pdf
Completed Sections	6
Course Links	https://www.linkedin.com/learning/programming-foundations-databases-2/?resume=false
Goal Sections	6
Quick note	https://www.linkedin.com/learning/certificates/49bdbcfc645fb4b58f9f87e0ca420b9544e6897bbd8d07b5f30e0 trk=share_certificate
Start and Finish Date	
Thumbnail	
Time	1h 25m 15s
Understanding %	
نسبة الإنتهاء	✅Done

Intro

▼ What is Database

- The make the work with a big amount of data more efficient.
- Provide structure for data
- Allow enforcement of rules for data
- Protect data from unauthorized access or changes
- A database is a structure that stores information in an organized , consistent, reliable, and searchable way.
- Spreadsheets
- 

- Structured data
- When you store data in a database, what is one advantage you get over a plain spreadsheet
  - Rows of data can be associated with each other across table
    - While some spreadsheets can simulate this capability with lookup functions, the ability to create relationships between tables is a core function of relational databases.



- The definition of how data in a database will be organized is called the `schema`.
  •The database's schema includes the information about the layout of tables and other information about the database itself.

Database Foundation

▼ Relational databases

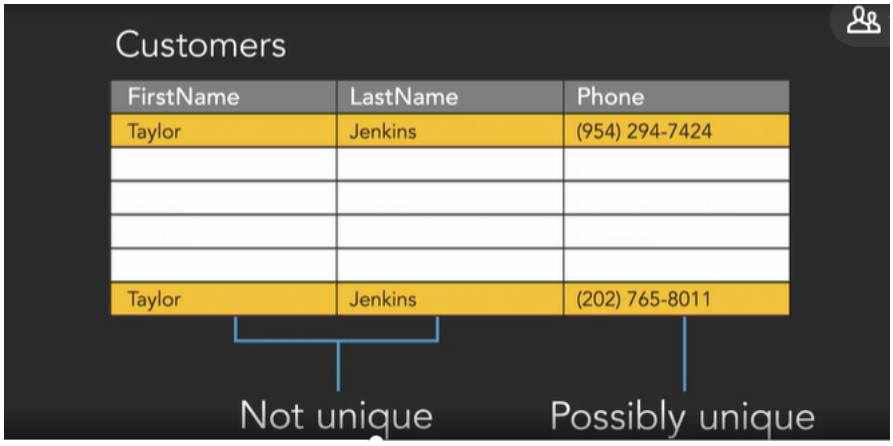
- In a relational database, tables are made up of rows, which represent instances of a given entity, and columns, which represent attributes of each entity.
  - Example
    - Customer = Entity

    - Dis=Entity

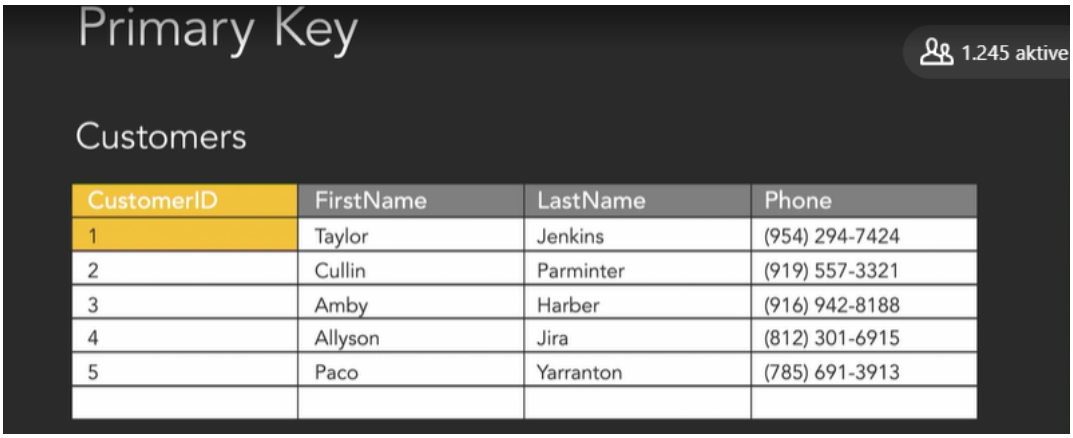
-

▼ Keys and unique values

- Unique values appear only once within a given column



  - Composite Key
    - Two or more fields taken together to act as a unique identifier
- Keys refer to only one record



  -

▼ Relationships

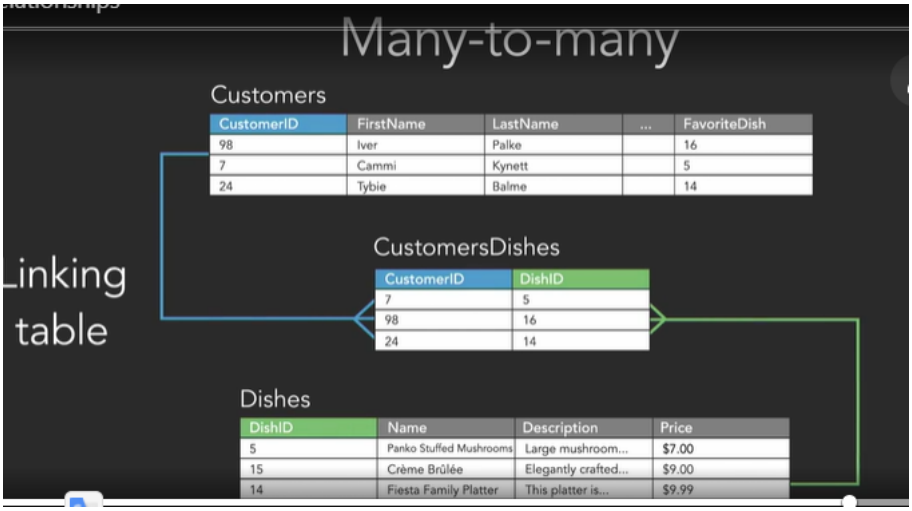
- If we need to tell the database that particular records should be associated with each other.
  1. One-to-many (most common)

a. This associates one record in one table with multiple records in another table.

i. 

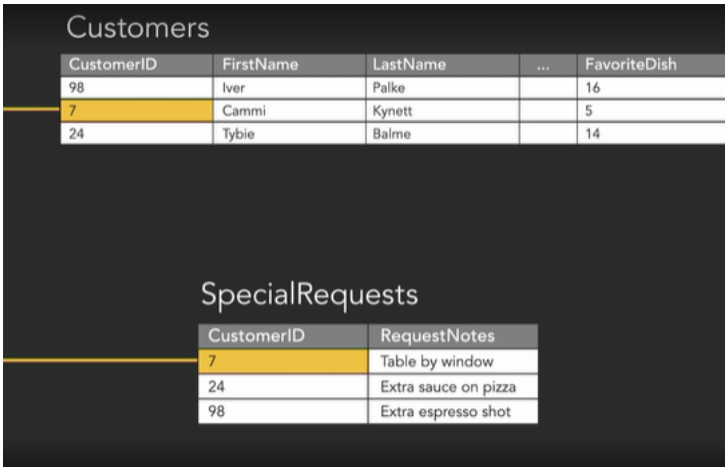
2. Many-to-many

a. we create a new table called an associative or linking table that contains columns for the foreign keys from the tables we're associating.



3. One-to-one

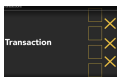
a. Unlike a one to many relationship, a one to one relationship associates only one record on one table **with only one record on another table,**



1.

▼ ACID and transactions

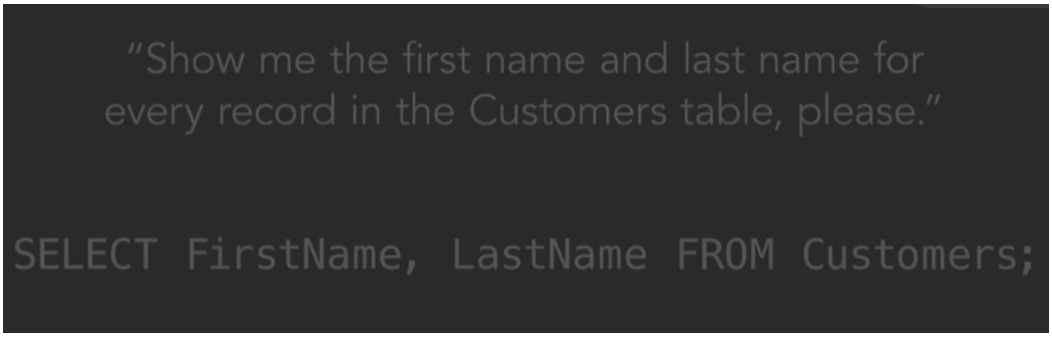
- the transaction have to be very accurate.
  - A transaction is a set of operations that must all be completed, and if for some reason any of the individual operations aren't completed, **no changes are made to the database.**



  - They have to be ACID
    - Atomic
      - there are indivisible, that pieces of it can’t be separated out.
    - Consistent
      - it means that whatever the transaction does, it needs to leave the database in a valid or consistent state. The actions in a transaction can't violate integrity rules that are defined for the database.
    - Isolation
      - means that while the activities in the transaction are being completed, nothing else can make changes to the data involved.
    - Durability
      - means that the information we change in the transaction is reported as complete, the data is there.

▼ Basic SQL

- is a database language
  - Structured
  - Query
  - Language
- Allows statements to be written for DBMS to interpret how to interact with data
- Offers feature to manage the database itself, such as creating of´r modifying tables and controlling access to tables
- Examples

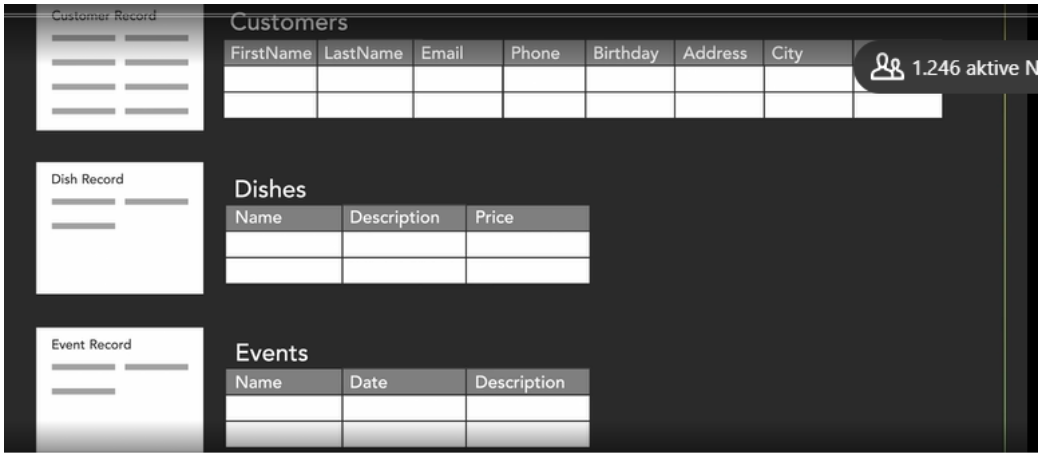


-

Tables

▼ Modeling and planning a database

- ER Diagram
  - A diagram that uses tables , fields, and relationships to plan a database.



  - Naming tables
    -

▼ Naming tables

- we have to name the table , with a name what descripe what includes this table
  - the column
    - we use the upperCamelCase , without spaces between the words

        - eg. FirstName
        - eg. LastName

▼ Columns and data types

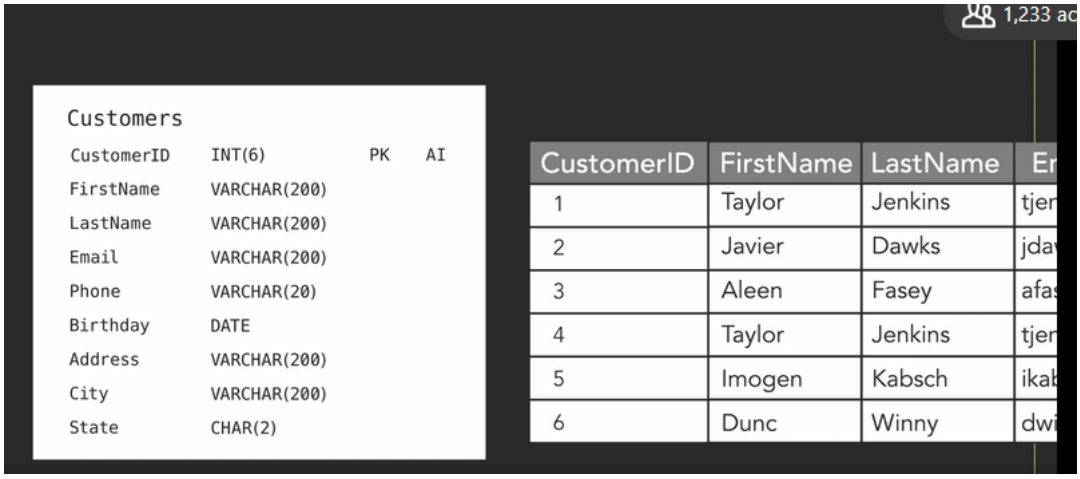
- Data types
    - Strings- alphanumeric characters and text
        - Char- fixed number of characters
        - VARCHAR- variable number of characters up to a maximum length
        - other types for longer text
    - Dates and times
        - Date 2019.03.09
        - DateTime 2019.03.09 16:51:00
        - TimeStamp (saved when record is updated)

▼ Numbers and other types

- Integers
- Double precision
- Floating point
- Decimals
- (Null) represents the absence of a value
    - if the column is empty
        - Null
    - if it is full
        - NOT NULL

▼ Primary and foreign key

- the primary key is a number what we add to the row to make the row special and don't repeat the same value



    - Composite key
        - A composite key is a combination of fields that uniquely identify a record.

Relationships

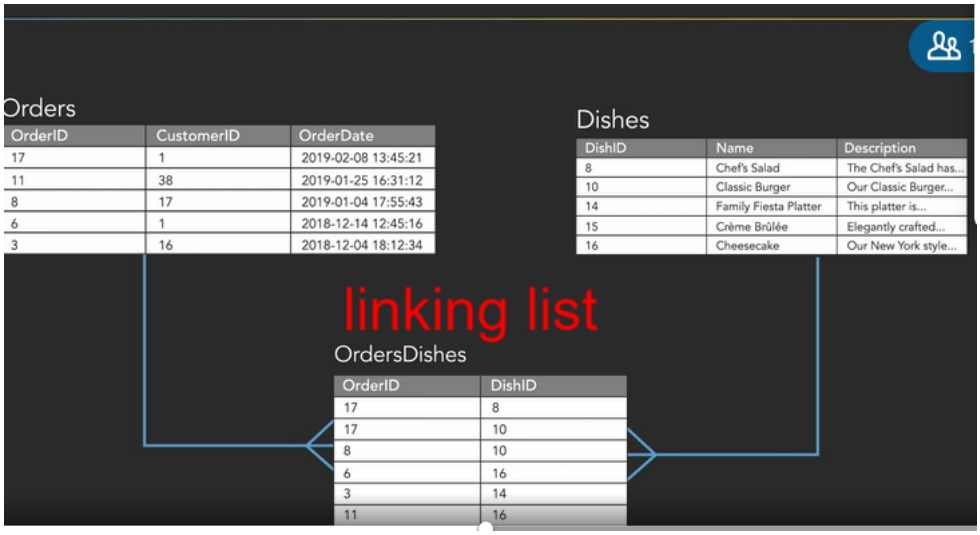
▼ Creating relationships

▼ Relationships

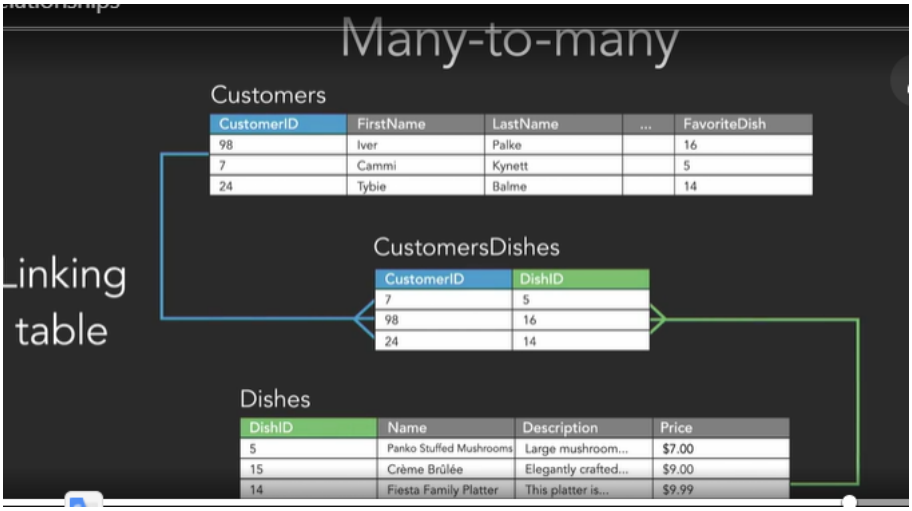
- If we need to tell the database that particular records should be associated with each other.
  1. One-to-many (most common)
     a. This associates one record in one table with multiple records in another table.
        i. Example

1.

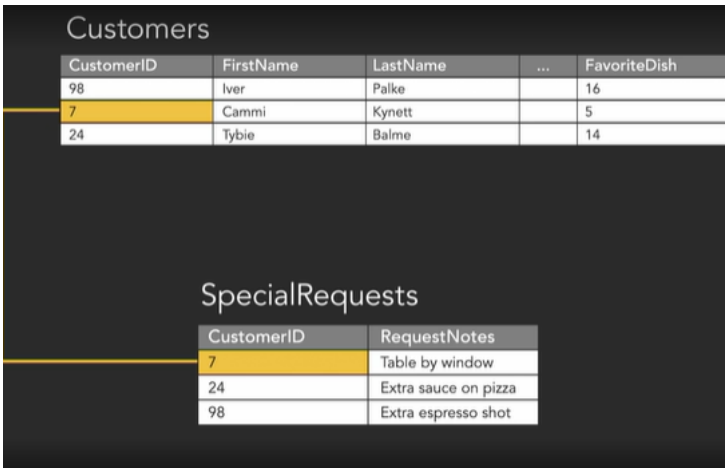
  2. Many-to-many
     a. we create a new table called an associative or linking table that contains columns for the foreign keys from the tables we're associating.
     b. Example



1.



3. One-to-one
    a. Unlike a one to many relationship, a one to one relationship associates only one record on one table **with only one record on another table,**
    b. it separate one row to be more secure



- Relationship rules and referential integrity
    - Databases are aware of relationships and won't allow a user to modify data in a way that violates those relationships.
    - What is it called if you delete a record and the database goes on and deletes other records associated with that record?
        - a cascading delete: If configured to do so, a delete operation can cascade across records linked with a relationship.

Database Optimization

▼ Normalization

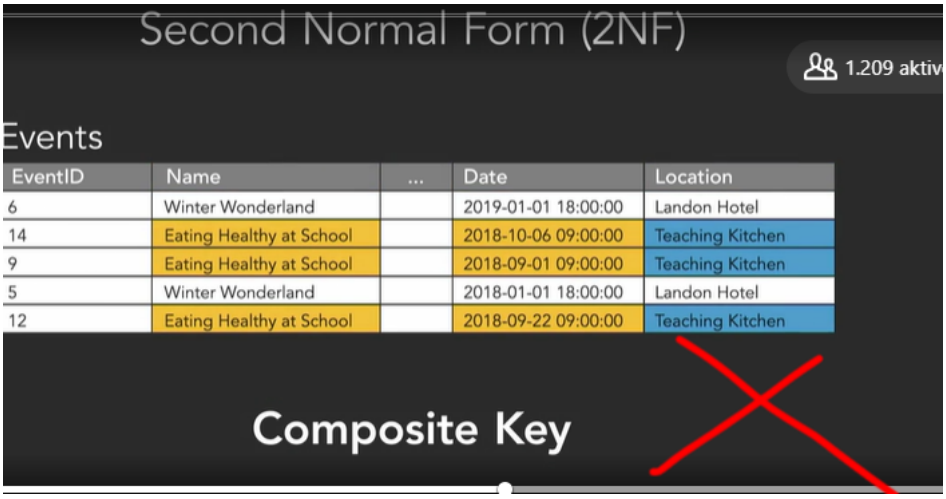
- There are 3 rules to otganizing data in a database
- When a normalization rule has been applied to a database, we can say that the database is in that normal form

▼ First normal form

- Values in each cell should be atomic and tables should have nonrepeting groups
  - Just one value in a cell
    -

▼ Second normal form

- No value in a table should depend on only part of a key that can be used to uniquely identify a row
  - this means that for every column in the table that isn't a key, each of the values must rely on only the whole key. The values must describe something about that row that we can't determine from just the part of a key
  - example



▼ Third normal form

- Values should not be stored if they can be calculated from another non-key field.

- Not only are we wasting resources by storing this information, but having it available here means that someone could edit the price in one column but not the other,

▼ Denormalization

- The process of intentionally duplicating information in a table , in violation of normalizaition rules
  - It's happen after normalization

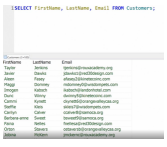
    - Denormalization is a trade-off. Gaining speed may reduce consistency.
  - If you can figure out the value of one non-key field in a row by looking at another non-key field in that same row, what do you violate?
    - Third Normal Form

Querying a Database

▼ In a Database we can:

- Creating tables
  - 

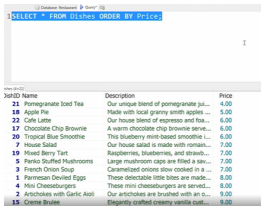
```
CREATE TABLE Customers (
CustomerID INT(6) NOT NULL AUTO_INCREMENT,
FirstName VARCHAR(200) NOT NULL,
LastName VARCHAR(200) NOT NULL,
Email VARCHAR(200)
PRIMARY KEY(CustomerID)
)
```

- Writing SQL queries (specific info what we ask for)
  - 

```
SELECT FirstName, LastName, Email FROM Customers;
```

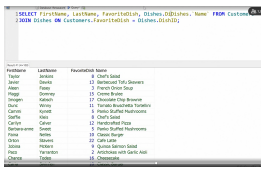
- Narrowing query results
  -

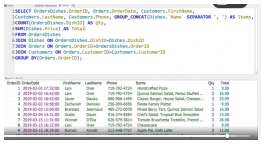
- Sorting results



- Aggregate functions

- Joining Tables





- Modifying data
  - Update

  - Insert

  - Delete

    -

▼ A SQL statement that returns requested records from the database is called:

a SQL query: All statements return a status when executed, but a query is a special case of statement that returns information you asked for.

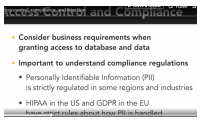
▼ Indexes, transactions, and stored procedures

- are all features offered by most DBMS tools.
  - Indexes
    - with an index on the fields the database stores a reference to what each value is in those fields and where it's located, and using that index, the database can **return information much faster , than scanning the whole database**
      - but when we add indexes, it increases the amount of time some operations, like inserting a record, will take.
  - Transactions
    - Transactions group queries or statements into a block of activities, where, if one of the components fails for any reason, the whole group of statements is not executed, **and anything that's partially done is rolled back.**

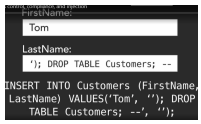
  - Stored procedure
    - is kind of like a program you write that's stored on the database server.
      - Are a series of commands stored on the database
      - Allow reuse of long or detailed queries instead of writing them for each use
      - Provide a safe way to deal with sensitive data

▼ Access control, compliance and injection#

- Access control
  - User accounts
  - table and column permission



- Sql injection
  - Type of attack that includes part of a Sql command entered as a value to hijack a query and change how it works
    -



▼ Softwares

Hadoop and Spark are often used for big data applications, database is often used in a big data context

Microsoft Access is generally considered a(n) `__desktop___` database platform.

Desktop databases are typically hosted on a workstation rather than a dedicated server, and they're designed to support a few to a few hundred users.