RMDS Workshop 2

Conversational SQL

Workshop Outline

Workshop Outline

What is SQL?

Workshop Outline

What is SQL?

What is a relational database?

Workshop Outline

What is SQL?

What is a relational database?

Normal Forms and Joining Tables

Workshop Outline

- What is SQL?
- What is a relational database?
- Normal Forms and Joining Tables
- SQL Verbs

Workshop Outline

What is SQL?

What is a relational database?

Normal Forms and Joining Tables

SQL Verbs

Hands-on SQL Practice

What is SQL?

What is SQL?

SQL stands for Structured Query Language

What is SQL?

SQL stands for Structured Query Language

This language is used to interact with relational databases

What is SQL?

SQL stands for Structured Query Language

This language is used to interact with relational databases

SQL is typically synonymous with databases in general

What are relational databases?

What are relational databases?

Relational databases are the most common type of database designed around a related set of tables

What are relational databases?

Relational databases are the most common type of database designed around a related set of tables

An example set of tables are customers, transactions, and items

What are relational databases?

- Relational databases are the most common type of database designed around a related set of tables
- An example set of tables are customers, transactions, and items
- Each of these tables are related, but each has information which would be redundant or unnecessary to other tables

What are relational databases?

Relational databases are the most common type of database designed around a related set of tables

An example set of tables are customers, transactions, and items

Each of these tables are related, but each has information which would be redundant or unnecessary to other tables

For example, we do not need all of a customer's available information to be included in the transcation table. We simply need the Customer Id refering us a customer within the customer table.

Normal Forms

Normal Forms

This reduction of redundancy by seperate of tables is known as database normalization.

Normal Forms

This reduction of redundancy by seperate of tables is known as database normalization.

Database normalization is designed to conserve database memory resources as the number of entries increases

Normal Forms

This reduction of redundancy by seperate of tables is known as database normalization.

Database normalization is designed to conserve database memory resources as the number of entries increases

It also alleviates confusion on which table is "ground truth" for a specific piece of information.

Joining Tables / Combining Data

Joining Tables / Combining Data

- Since the data are normalized into seperate tables, one of the most important "SQL verbs" is Join

Joining Tables / Combining Data

Since the data are normalized into seperate tables, one of the most important "SQL verbs" is Join

In order to join two database tables, a primary key is required which uniquely matches which rows from each table should correspond.

Joining Tables / Combining Data

Since the data are normalized into seperate tables, one of the most important "SQL verbs" is Join

In order to join two database tables, a primary key is required which uniquely matches which rows from each table should correspond.

As such, it is typical practice for each table to have an "ID" data column as the first column. This is used internal to the database to ensure uniqueness regardless of the data.

Hands-on Exericses: Getting Started

Hands-on Exericses: Getting Started

1. To start, navigate to sqlfiddle.com

Hands-on Exericses: Getting Started

1. To start, navigate to sqlfiddle.com
2. Go to our workshop repository, and copy the database schema

Hands-on Exericses: Getting Started

1. To start, navigate to sqlfiddle.com
2. Go to our workshop repository, and copy the database schema
3. Enter SQL on the right-hand side to query the example database

SQL Verbs: Viewing Data

```
SELECT *
FROM customers;
```

SQL Verbs: Viewing Specific Data Columns

```sql
SELECT id, customer_name
FROM customers;
```

SQL Verbs: Joining Tables

```
SELECT
customers.name, transactions.transaction_date
FROM customers
JOIN transactions
ON customers.id = transactions.customer_id;
```

SQL Verbs: Different Types of Joins

```
SELECT *
FROM customers
LEFT JOIN transactions
ON customers.id = transactions.customer_id;
```

SQL Verbs: Sorting Data

```
SELECT *
FROM transactions
ORDER BY transaction_date;
```

SQL Aggregate Functions

```sql
SELECT COUNT(*)
FROM customers;
```

List of SQL Aggregate Functions

COUNT() - Counts the number of rows

AVG() - Takes an average over the rows

SUM() - Takes a sum over the the rows

MIN() - Takes the minimum over the rows

MAX()- Takes the maximum over the rows

Grouping Data

```
SELECT customer_id, COUNT(*)
FROM transactions
GROUP BY customer_id;
```

Filtering Data

```
SELECT name, state
FROM customers
WHERE state = 'Montana';
```

Filtering Data (Multiple Conditions)

```
SELECT name, state
FROM customers
WHERE state = 'Montana' OR
state = 'Illinois';
```

Filtering Data (Multiple Conditions)

```
SELECT name, state
FROM customers
WHERE state = 'Montana' AND
id < 3;
```

Tips for Exercises (and SQL in general)

Tips for Exercises (and SQL in general)

Start with a basic query

Tips for Exercises (and SQL in general)

Start with a basic query

Slowly add data and transformations

Tips for Exercises (and SQL in general)

Start with a basic query

Slowly add data and transformations

Run the query often

Hands-on Exericses

1. Write a SQL Query to compute number of transactions and total amount spent per customer
2. Write a SQL Query to find out who purchased first (in regards to transaction_date)
3. Write a SQL Query to find the number of transactions per Day
4. Write a SQL Query which finds the customer who has spent the most
5. Write a SQL Query which finds the customer who has spent the least
6. Write a SQL Query which determines the average amount spent per day