COURSEWARE

Professional Skills

Agile Fundamentals

Jira

Git

Databases Introduction

- ○ Installing MySQL on Windows
- ○ Provision a MySQL Server (Google Cloud Platform)
- ○ Introduction to Relational Databases
- ○ Data Design
- ○ Data Definition Language (DDL)
- ○ Entity-Relationship Diagrams
- ○ Data Manipulation Language (DML)
- ○ Data Query Language using SELECT
- ○ Aggregate Functions
- ○ Nested Queries
- ○ Joins
- ○ Data Normalisation

Java Beginner

Maven

Testing (Foundation)

Java Intermediate

HTML

CSS

Javascript

Spring Boot

Selenium

Sonarqube

Advanced Testing (Theory)

Cucumber

MongoDB

Express

NodeJS

Aggregate Functions

Contents

Overview

An aggregate function or aggregation function is a function where the values of multiple rows are grouped together to form a single summary value. There are a number of aggregate functions supported in MySQL, all of which aim to get a single result from several records:

- `COUNT` – counts the number of fields
- `SUM` – gets the sum total of a field
- `MIN/MAX/AVG` – gets the minimum/maximum/average value from a field

COUNT

The `COUNT()` function returns the number of records that matches a specified criteria:

```sql
SELECT COUNT(field_name1) FROM table_name;
```

For example, if we wanted to count the number of orders placed:

```sql
SELECT COUNT(order_id) FROM orders;
```

SUM

The `SUM()` function returns a calculated total from a numeric field:

```sql
SELECT SUM(field_name1) FROM table_name;
```

For example, if we wanted to count the number of items in the order line:

```sql
SELECT SUM(quantity) FROM order_line;
```

MIN, MAX, and AVG

The `MIN()` function returns the smallest value in a field:

```sql
SELECT MIN(price) FROM product;
```

The `MAX()` function returns the largest value in a field:

```sql
SELECT MAX(price) FROM product;
```

The `AVG()` function returns the average value in a field:

```sql
SELECT AVG(price) FROM product;
```

React

Express-Testing

Networking

Security

Cloud Fundamentals

AWS Foundations

AWS Intermediate

Linux

DevOps

Jenkins Introduction

Jenkins Pipeline

Markdown

IDE Cheatsheet

GROUP BY

The `GROUP BY` command is often used in conjunction with aggregate functions to bring together sections of data based on common values within a field.

For instance, if you wanted to find the most expensive order made by each customer, we could try something like this:

```
SELECT customer_id, MAX(total) AS max_total FROM orders GROUP BY customer_id;
```

Tutorial

There is no tutorial for this module.

Exercises

Design, write and run `SELECT` queries for your games shop database that will answer typical questions you would ask of a retail database (price of the most expensive item, most common customer, etc.)

(*note: if you have not tested `SELECT` statements yet, refer back to the Data Query Language module for context*)

▶ Solution