

## MANAGING VIEWS

**CREATE VIEW** `simple_view`  
**AS SELECT** name, age **FROM** employees;

Create a view that consists of name and age columns from the employees table.

**CREATE VIEW** `active_emp`  
**AS SELECT** name, age **FROM** employees  
**WHERE** status = 'active';  
**WITH CASCADED CHECK OPTION;**

Create a view that shows active employees only and use the **CASCADED CHECK OPTION**.

**CREATE RECURSIVE VIEW** `example_view`  
**AS**  
**SELECT** 1  
**UNION ALL**  
**FROM** example\_view **WHERE** n < 10;

Create a recursive view that generates numbers from 1 to 9.

**CREATE TEMPORARY VIEW** `temp_view`  
**AS**  
**SELECT** 1  
**FROM** example\_view **WHERE** n < 10;  
Create a temporary view.

**DROP VIEW** `simple_view`;  
Delete the `simple_view`.

## MANAGING INDEXES

**CREATE INDEX** `idx_name`  
**ON** employees (**last\_name**)

Create an index on the last\_name column of the employees table.

**CREATE UNIQUE INDEX** `unique_email`  
**ON** users (email)

Create a unique index on the email column of the users table.

**DROP INDEX** `idx_name` **ON** employees.

## SQL AGGREGATE FUNCTIONS

**AVG** returns the average:  
**SELECT** AVG(salary) **FROM** employees;

**COUNT** returns the count:  
**SELECT** COUNT(\*) **FROM** orders;

**SUM** returns the total:  
**SELECT** SUM(amount) **FROM** sales;

**MAX** returns the maximum value:  
**SELECT** MAX(price) **FROM** products;

**MIN** returns the minimum value:  
**SELECT** MIN(stock) **FROM** inventory;

## ROLES AND PERMISSIONS (MySQL)

### CREATING USERS

**CREATE USER** 'alice'@'localhost'  
**IDENTIFIED BY** 'password123';

Create a new user 'alice' with a password "password123" that can connect from localhost.

### CREATING ROLES

**CREATE ROLE** `manager`;  
Create a new role called `manager`.

### ASSIGNING ROLES AND GRANTING PERMISSIONS

**GRANT** **SELECT, INSERT**  
**ON** company\_db.\* **TO** 'company\_db.\* **TO** 'alice'@localhost;  
Grant SELECT and INSERT privileges on all tables in company\_db to the user `alice` connecting from localhost.

**GRANT** `manager` **TO** 'bob'@localhost;  
Assign the manager role to the user `bob` connecting from localhost.

### REVOKING ROLES AND PERMISSIONS

**REVOKE** **SELECT, INSERT** **ON** company\_db.\*  
**FROM** 'alice'@localhost; **REVOKE** `manager` **FROM** 'bob'@localhost;

**REVOKE** `manager` role from the user `bob`

### DROPPING ROLES

**DROP ROLE** `manager`;  
Delete the manager role.