IT240 Database Administrations

Activity2: SQL statements

Objectives: develop understanding of SQL and structure.

To run a query, make sure to select the query, and press on the 'Execute' button or use the keyboard shortcut (e.g., Ctrl + Enter) to initiate the query execution, otherwise your query will not be executed.

**Step 1: Creating and deleting the database:**

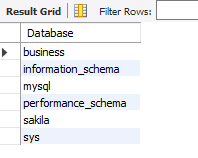
1. Open MySQL Workbench, click on **root** and enter the root password.

* To display a list of all databases available in the database system we use the command:

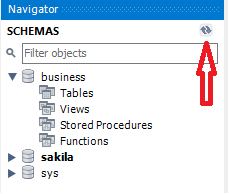
**SHOW databases;**

1. Write an SQL statement to list all available databases on your database system:
   * Make sure to execute the command by selecting it then clicking on the  button;

* Notice the result in the results grid



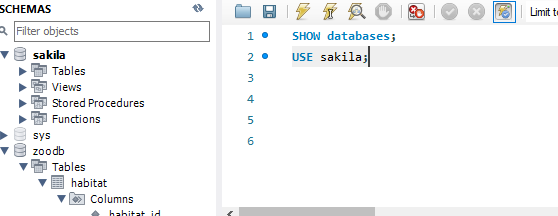
* Refresh the navigation bar to see the changes in the navigation bar.



* To select or switch to a specific database, we use the following command:

**USE databaseName;**

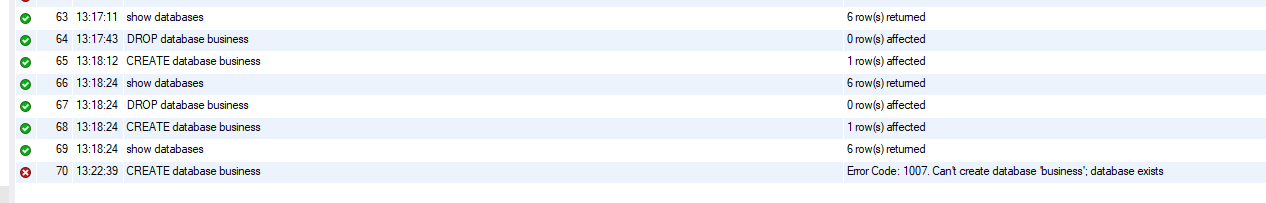
1. Write an SQL statement to select **sakila** database.
   * After exciting the query **sakila** in the navigation bar will be highlighted, indicating that we are currently using this database.



* To create a new database, we use:

**CREATE DATABASE DatabaseName;**

1. Write an SQL statement to create a new database, name it ***business***.
2. Write an SQL statement to switch to the ***business***.
3. Write SQL statement to create another database with the name ***business.*** What happens?
   * Each database is uniquely named.
   * Notice the error in the output bar:



* To delete a database from your database system we use the command:

**DROP DATABASE DatabaseName;**

1. Write an SQL statement to delete the database we created last week ***zoodb***.
   * If you missed last Thursday class create a zoodb and delete it.

**Step 2: Tables and Columns**

* To create a table, we use the following command:

**CREATE TABLE** TableName (

Column1 **DATA\_TYPE**,

Column2 **DATA\_TYPE**,

...

ColumnN DATA\_TYPE );

* MySQL has the following datatypes:



1. Write an SQL statement to switch to the ***business*** database.
2. Write an SQL statement to create an ***employee*** table in the ***business*** database, your table should include the following columns, make sure to use the appropriate datatype:

*employee\_id*: The company has 150 employees, the first employee will get id 1 and so on.

*employee\_name* : Variable-length string with maximum 40 characters.

employee\_date: Year, month, and day.

*Gender*: variable-length string with maximum 1 character.

*department\_id:* the company has only 20 departments, numbered from 1--20

*last*\_date\_modified: Year, month, day and time.

Hint: gender is only one character long, so we can use CHAR(1), what is the other column that might take same datatype?

* To add, delete, or modify columns in an existing table, we use the command:

**ALTER TABLE;**

* To add new column to the table we use:

**ALTER TABLE** TableName

**ADD** columnName dataType;

* To modify existing column:

**ALTER TABLE** TableName

**CHANGE** currentColumnName NewColumnName newDataType;

* To delete a column:

**ALTER TABLE** TableName

**DROP** ColumnName;

1. Write an SQL statement to modify **employee** table by adding new column ***Salary***, make sure to select the most appropriate datatype.
2. Write an SQL statement to change the name of the column ***last*\_date\_modified** to **mod\_time\_stamp,** don’t change its datatype**.**
3. Write an SQL statement to delete the ***gender*** column.
4. Write an SQL statement to add a column named ***employee\_date*** with same datatype you used above in step 9. What happened? Read the error in the output bar.

* Tables must obey relational rules, including:
  + Exactly one value per cell
    - Unknown data is represented with a special NULL value.
  + No duplicate column names
    - Duplicate column names are allowed in different tables, but not in the same table.
  + No duplicate rows
    - No two rows may have identical values in all columns.
  + No row order(**data independence**)
    - The organization of rows on a storage device, such as a disk drive, never affects query results.

1. Write an SQL statement to create another table name it ***department***, your table should have the following columns:
   * dep\_id: it should take a number from 1- 20
   * dep\_name: string, maximum length 40.

**Step 3: inserting values to the tables:**

* We use select statement to insert values in a table, this is how looks like:

**INSERT INTO** TableName (

Column1, Column2,

...

)

**VALUES** (Value1), (Value2),

...

);

* + Important: any value that has VARCHAR, CHAR, date or DATETIME datatype has to be enclosed in single or double quotes.

1. Write an SQL statement to insert three different rows in the ***employee*** table.
2. Write an SQL statement to insert three different values to the ***department*** table.

**Step 4: Query the data**

* Select with asterisk is used to select all elements from specific table, here is how it looks like:

**SELECT** \* **FROM** TableName;

1. Write an SQL statement that selects all records in the ***employee*** table.

* Select with column names is used to retrieve specific columns from a table in the database. Here is how it looks like:

**SELECT** Column1, Column2, ... **FROM** TableName;

1. Write an SQL statement that selects only ***emplyee\_id***, ***employee\_name*** and ***salary*** from the ***employee*** table.
   * A select statement with expression is used to retrieve data from table, but instead of selecting specific columns, you can use expressions to perform calculations or transformation on the data. Here is how it looks like:

**SELECT** Expression1, Expression2, ... **FROM** TableName;

1. Write an SQL statement that retrieves ***employee\_name*** from table ***employee*** and calculate an increasing salary (5% raise) using the following expression: salary \* 1.05 and aliasing it as ***increase\_salary***.

* **Hint**: this is an example of select statement with expression:
  + SELECT name, id, grade + 5 AS new\_grade FROM student;
  + Notice that ***increase\_salary*** calculated in the select statement will not be stored in the employee table, the results for this calculation is temporary and exists only for the duration of query execution.
* A SELECT statement has an optional ***WHERE*** clause that specifies a condition for selecting rows.
  + A row is selected when the condition is TRUE for the row values.
  + A row is omitted when the condition is either FALSE or NULL.
* Here is how it looks like:

**SELECT** Expression1, Expression2, ... **FROM** TableName WHERE Condition;

1. Write an SQL statement that selects ***emplyee\_id***, ***employee\_name***, and ***salary*** from the ***employee*** table only for employees who works in department 1.