

# MySQL

# Contents

Install and Setup MySQL .....	3
About Database .....	6
Create a database for Coffee Shop menu .....	8
Answer few queries related to the menu .....	12
Managing various sizes of the drink .....	16
Aggregating rows using Group By .....	19
Display Menu Items with Size.....	20
Find absentees using outer join .....	25
Create stored procedure to insert data .....	27
Create stored function to display ABSENT instead of null .....	31
Record all INSERT, UPDATE and DELETE operations on a table using triggers.....	33
Design database for coffee shop .....	36
Create the database design for managing Movie information .....	<b>Error! Bookmark not defined.</b>
Miscellaneous Topics .....	42

## Install and Setup MySQL

### Download MySQL

- ❖ In browser go to <https://dev.mysql.com/downloads/mysql/>
- ❖ Click the download button adjacent to "Windows (x86, 64-bit) ZIP Archive"
- ❖ Extract the root folder "mysql-8.0.20-winx64" into D:

### Setup Database

- ❖ Create new folder named "data" in D:\mysql-8.0.20-winx64
- ❖ In windows file explorer go to D:\mysql-8.0.20-winx64\bin folder
- ❖ Click in the empty space next to "bin" folder on the top of the window.
- ❖ Type "cmd" and press enter.
- ❖ This open command prompt in the bin folder of mysql.
- ❖ Execute the following command:

```
mysqld --initialize --console --basedir=D:\mysql-8.0.20-winx64 --datadir=D:\mysql-8.0.20-winx64\data
```

**IMPORTANT NOTE:** If the folder or version number is different, then ensure appropriate folder name.

- ❖ Make note of the password generated in the console, look for the line below:  
**A temporary password is generated for root@localhost: YQU6E,xraigx**
- ❖ Select the password and press enter.
- ❖ Open Notepad and press Ctrl+V which will paste the password.
- ❖ If the MySQL server installation fails or gives error for some reasons, the use the below link to download MySQL server zip file:  
<https://drive.google.com/file/d/1CRRJ-xUtDapgFIZPK2dJ3qh1IHJCeLH6/view?usp=sharing>

### Change password for 'root' user

- ❖ Now execute the following command to start the server.

```
mysqld --console
```

- ❖ Wait till it displays the message "ready for connections"
- ❖ Open a new command prompt window in mysql\bin folder.
- ❖ Run the following command:

```
mysql -u root -p
```

- ❖ Copy the password from notepad
- ❖ Open the command prompt window and right click, which will paste the password.
- ❖ Press enter and providing the password.
- ❖ It will open a mysql> prompt
- ❖ Execute the following command:

```
alter user 'root'@'localhost' identified by 'password';
```

- ❖ Execute the following command and press enter

```
exit
```

- ❖ Run the following command:

```
mysql -u root -p
```

- ❖ Enter the password as "password"
- ❖ It should not login in the new password.

### Stopping MySQL Server

- ❖ Open the command window where the server is running
- ❖ Press Ctrl+C to stop the server.

### Starting MySQL Server and MySQL Client

- ❖ Open command prompt in mysql bin folder.
- ❖ Execute the following command to start the server:

```
mysqld --console
```

- ❖ Open another command prompt in mysql bin folder.
- ❖ Execute the following to start the client. Enter 'password' as password when prompted.

```
mysqld -u root -p
```

### Install MySQL Workbench

- ❖ Open link <https://dev.mysql.com/downloads/workbench/> in browser
- ❖ Click "Download" link next to "Windows (x86, 64-bit), MSI Installer"
- ❖ Click "No thanks, just start my download" link
- ❖ A file with extension .msi will be downloaded
- ❖ Run the downloaded MSI file and follow the instructions to install.
- ❖ Instructions for 8.0 installation:
  - Click "Next"
  - Change installation folder if required and then click "Next"
  - Select "Complete" for "Setup Type" and then click "Next"
  - Click "Install"

### Test MySQL Workbench connection with MySQL Server

- ❖ Ensure that MySQL Server is running and launch MySQL Workbench
- ❖ Enter "Connection Name" as "local" and click "Test Connection" button

❖ Enter password as "password" and check if the test connection is successful

## About Database

### **Contact Details (Understanding the need for a database)**

- ❖ Is it possible remember all the contact numbers of our mobile phone?
- ❖ How does one find a contact in a phone?
- ❖ Where are the contact details stored?
- ❖ When a new phone is bought, how we get the contact details from the old phone?
- ❖ If the phone does not have storage, where possibly can we store the contact details?

### **Non-Electronic storage mediums**

- ❖ Paper
- ❖ Notebook
- ❖ Phonebook
- ❖ Sticky Notes

### **Primary Memory**

- ❖ Main Memory of a computer
- ❖ RAM, ROM are examples for Primary Memory
- ❖ Primary Memory is volatile
- ❖ When typing content in Notepad or MS Word, it is stored in Primary Memory
- ❖ Data in Primary Memory is lost if power is lost

### **Secondary Memory**

- ❖ Stores high volume of data
- ❖ Slower than Primary Memory (means that it is take more time to read and write)
- ❖ Examples are Floppy Disks, Hard Disks, Flash/SSD, CD, DVD, Blue Ray, USB Drives and Magnetic Tapes
- ❖ When saving a file in Notepad or MS Word, it is stored in a secondary storage device.

### **Enterprise Data**

- ❖ How does an airline manage their data about flights, schedules, booked tickets, passenger details, boarding card details, cancellations, etc.? For example, Emirates manages approximately 56 million passengers per year.
- ❖ How does a telecom company manage their data about their customers, call data, plans, etc.?

### **Evolution of Database**

- ❖ Three eras of database technology:

Navigational

SQL/Relational

Post-relational

- Navigational (IBM's Information Management System (IMS) on System/360)
- SQL/Relational
  - **Edgar F. Codd** working in IBM was not happy with IMS
  - In 1970, he wrote a number of papers on relational database
  - Codd defined **13 rules** for defining a relational database
  - There isn't any product that follows all 13 rules. Oracle follows eight and half rules out of the 13 rules.
  - In short, this is called as **RDBMS** (Relational Database Management System)
  - Multiple commercial products got rolled out during late 1970s
    - IBM developed SQL/DS, later called as **DB2**
    - **Larry Ellison** developed **Oracle** V1 in 1978 and Version 2 in 1979
    - **Stonebraker** developed **Postgres**
  - During 1976, another model called **entity-relationship** model evolved for **database design**
  - During the age of desktop computing in 1980s, the desktop computers were empowered with managing data using spreadsheets like Lotus 1-2-3 and database software like dBase.
  - During 1990s, due to rise in Object Oriented Programming, there was a need to represent data as objects, which resulted in the evolution of Object Relational Mapping (ORM) libraries. Hibernate and iBatis are example for ORM.
- Post-relational (During 2000)
  - XML Database
  - NoSQL – Cassandra, Apache CouchDB, MongoDB, Apache Ignite, Neo4J

## Create a database for Coffee Shop menu

### Problem

Create a database to store a Coffee Shop menu item details. Refer menu details below.

Menu Item	Price
CAFFEE LATTE	\$ 2.70
CAPPUCCINO	\$ 2.70
CAFFEE MOCHA	\$ 3.00
ESPRESSO	\$ 1.55
ESPRESSO MACCHIATO	\$ 1.60

### Solution

- ❖ Start MySQL server if it is not running
- ❖ Open MySQL client in command prompt
- ❖ Execute the following statement to create the database.

```
mysql> create schema learndb;
```

- ❖ The execute the following command to switch to the newly created database.

```
mysql> use learndb;
```

- ❖ Execute statement to create table for storing menu items.

```
mysql> create table menu(item varchar(30), price decimal(5,2));
```

- ❖ Insert a menu item using the statement below:

```
mysql> insert into menu (item, price) values ('CAFFEE LATTE', 2.70);
```

- ❖ Use the statement below to see the data available in the menu table

```
mysql> select * from menu;
```

- ❖ The output should look something like this:

```
+-----+-----+
| item      | price |
+-----+-----+
| CAFFEE LATTE | 2.70 |
+-----+-----+
```

- ❖ Go to MySQL command window
- ❖ Use up arrow key to get the previous insert statement
- ❖ Modify the data with the next menu item values and insert all menu item details.
- ❖ Finally run the select query to check the data.

```
mysql> select * from menu;
+-----+-----+
| item      | price |
+-----+-----+
| CAFFEE LATTE      | 2.70 |
| CAPPUCCINO        | 2.70 |
| CAFFEE MOCHA      | 3.00 |
| ESPRESSO          | 1.55 |
| ESPRESSO MACCHIATO | 1.60 |
+-----+-----+
```



## Explanation of executed SQL Commands

### ❖ Understanding the schema creation:

```
mysql> create schema learndb;
```

- 'CREATE SCHEMA' creates a new database. A schema will contain a set of tables and database objects.
- The 'use' command is MySQL client specific command to switch between multiple databases within a database server.
- Execute the drop command to understand how to remove a database.

```
mysql> drop schema learndb;
```

- Execute create command again to create the database. Use up arrow to get the previous commands.

### ❖ Understanding the create table command:

```
mysql> create table menu(item varchar(30), price decimal(5,2));
```

- 'create table' command helps to create a new table.
- A table is a collection of related data entries
- A table consists of columns and rows
- 'menu' defines the table name
- 'item' and 'price' are columns in the table
- 'varchar' and 'decimal' are datatypes.
- 'varchar(30)' defines that 30 characters can be stored in this column
- 'decimal(5,2)' defines numerical values that can be stored with decimal place. In this definition it defines to store a maximum of 5 digits with 2 digits for decimal places.
- Execute the drop command to check how to remove a table from the database.

```
mysql> drop table menu;
```

- Use up arrow to run the create table command again.
- Few data types listed for reference:
  - Numeric - INT, DECIMAL, FLOAT, DOUBLE
  - String – CHAR, VARCHAR (CHAR is used for fixed width)
  - BLOB – Binary large objects (images files, documents, etc.)
  - LONGTEXT – Large non-binary string
  - Date and Time – DATE, TIME, DATETIME
- Refer data types in this link <https://www.mysqltutorial.org/mysql-data-types.aspx/>

### ❖ Understanding the insert command:

```
mysql> insert into menu (item, price) values ('CAFFEE LATTE', 2.70);
```

- 'insert' command adds data into table
- 'menu' refers to the table in which data needs to be inserted.
- The '(item, price)' defines the columns in which the values are supplied.
- The brackets followed after 'values' specifies the actual data values.
- The column names order defined and the values should match.

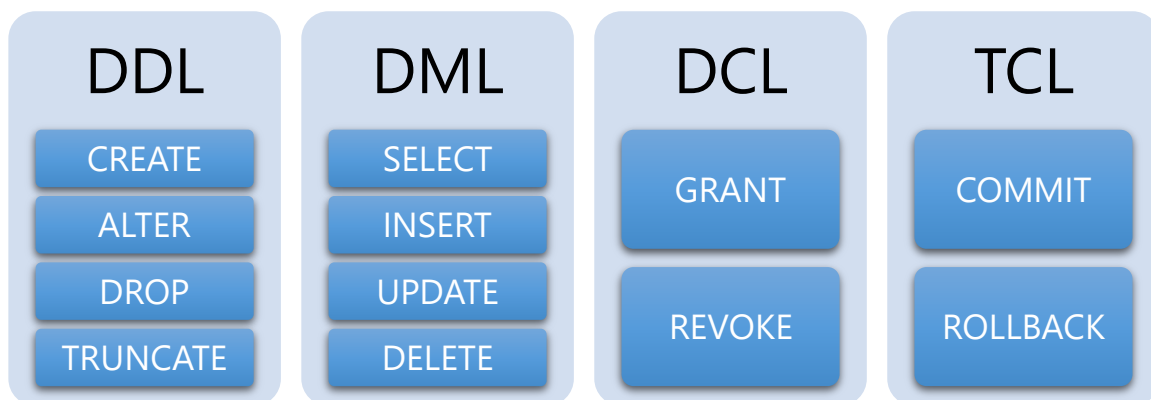
❖ Understanding the select command:

```
mysql> select * from menu;
```

- This command is used to view the content of a table
- The columns displayed are ordered in the order in which the columns are defined when creating the table.
- Instead of '\*' the respective column names can be included in a custom order.
- The table name to be read is defined after the 'from' clause.
- There is not guarantee that the result will be shown in the order how we inserted the data
- Use the 'show tables' command to view the list of tables in a database. This command is specific to MySQL client.

### Explanation (SQL Commands classification)

❖ SQL Commands are classified under 4 categories



❖ **Data Definition Language (DDL)** - Command that helps in creating, altering and removing database objects like schema, table, index, views, stored procedure, function and triggers.

- CREATE – Creates objects
- ALTER – Modifies already created objects
- DROP – Deletes objects from database
- TRUNCATE – Removes all records from table

❖ **Data Manipulation Language (DML)** - Commands that helps to add, modify and delete data in the database.

- SELECT – Read data from database
- INSERT – Add data into table
- UPDATE – Update existing data in table
- DELETE – Remove existing data in table

❖ **Data Control Language (DCL)** - Commands that help in managing access to the database.

- GRANT – Provide access to database objects
- REVOKE – Withdraw access to database objects

❖ **Transaction Control Language (TCL)** - Commands to manage transaction. Grouping a set of transactions as a single unit is called a transaction. Commands executed immediately updates the database, executing them as transactions helps to group a set of commands and save them one all transactions are done.

- COMMIT – Save all the transactions done so far
- ROLLBACK – Revert back all transactions done so far and retain the previous state.

### Check your understanding

- ❖ What is the purpose of a database?
- ❖ List the three eras of database technology
- ❖ What are the four classifications of database commands?
- ❖ What is DDL? What kind of commands come under this category?
- ❖ What is the purpose of 'create' command?
- ❖ What is the purpose of 'drop' command?
- ❖ What is DML? What kind of commands come under this category?
- ❖ What is the purpose of select command?
- ❖ What is the purpose of insert command?
- ❖ What is DCL? What kind of commands come under this category?
- ❖ What is TCL? What kind of commands come under this category?
- ❖ What is the syntax to create a new schema?
- ❖ How to remove an existing schema?
- ❖ What is the command in MySQL to switch between databases?
- ❖ What is the syntax of create statement for creation of a new table?
- ❖ What is the syntax of insert statement for adding a new row into a table?
- ❖ What is the syntax of select statement to view rows of a table?
- ❖ What does the asterisks in select statement represent?

## Answer few queries related to the menu

### Problem

Below are few queries from customers regarding the drinks in the coffee shop. Write appropriate SQL queries to get the result.

- ❖ What is the price of Caffee Mocha?
- ❖ What drinks comes at the price of 2.70 dollars?
- ❖ I want to have a drink which is less than two dollars. What are the options available?
- ❖ I want to see the drinks in increasing price order.
- ❖ I want to see the drinks in decreasing price order.
- ❖ I want to see the drinks sorted alphabetically.
- ❖ I want to see the drinks sorted in reverse alphabetic order.
- ❖ What is the maximum price in the menu?
- ❖ What is the minimum price in the menu?
- ❖ What items comes under the maximum price?
- ❖ What items comes under the minimum price?
- ❖ What items are available under 'ESPRESSO'?
- ❖ List all items between 2 to 3 dollars.
- ❖ List all items that starts with 'ESPRESSO' or 'CAFFEE'
- ❖ List all items that does not start with 'CAFFEE'

### Solution

Find below the SQL statement for each one of the below queries:

- ❖ What is the price of Caffee Mocha?  

```
mysql> select * from menu where item = 'CAFFEE MOCHA';
```
- ❖ What drinks comes at the price of 2.70 dollars?  

```
mysql> select * from menu where price = 2.7;
```
- ❖ I want to have a drink which is less than two dollars. What are the options available?  

```
mysql> select * from menu where price < 2;
```
- ❖ I want to see the drinks in increasing price order.  

```
mysql> select * from menu order by price;
```
- ❖ I want to see the drinks in decreasing price order.  

```
mysql> select * from menu order by price desc;
```
- ❖ I want to see the drinks sorted alphabetically.  

```
mysql> select * from menu order by item;
```
- ❖ I want to see the drinks sorted in reverse alphabetic order.  

```
mysql> select * from menu order by item desc;
```
- ❖ What is the maximum price in the menu?

```
mysql> select max(price) from menu;
```

- ❖ What is the minimum price in the menu?

```
mysql> select min(price) from menu;
```

- ❖ What items comes under the maximum price?

```
mysql> select * from menu where price = (select max(price) from menu);
```

- ❖ What items comes under the minimum price?

```
mysql> select * from menu where price = (select min(price) from menu);
```

- ❖ What items are available under 'ESPRESSO'?

```
mysql> select * from menu where item like '%ESPRESSO%';
```

- ❖ List all items between 2 to 3 dollars.

```
mysql> select * from menu where price > 2 and price < 3;
```

Alternate query, this includes the range provided.

```
mysql> select * from menu where price between 2 and 3;
```

- ❖ List all items that starts with 'ESPRESSO' or 'CAFFEE'

```
mysql> select * from menu where item like 'ESPRESSO%' or item like 'CAFFE%';
```

- ❖ List all items that does not start with 'CAFFEE'

```
mysql> select * from menu where item not like 'CAFFE%';
```

## Explanation

### SQL

- ❖ SQL is the standard language for accessing and manipulating databases.
- ❖ SQL stands for Structured Query Language
- ❖ SQL is defined using the ANSI standard
- ❖ SQL keywords are not case sensitive

### SELECT

- ❖ The SELECT statement is used to retrieve data from database. Refer syntax below:

```
SELECT column1, column2, column3, ..  
FROM table-name
```

- ❖ Asterisks can be used to represent all columns

### WHERE

- ❖ The WHERE clause is used to filter records. Find the syntax below:

```
SELECT column1, column2, column3, ..  
FROM table-name  
WHERE condition
```

- ❖ The WHERE clause retrieve records based on a condition

- ❖ WHERE clause can also be applied to UPDATE and DELETE statements. The update or delete happens on the rows where the condition satisfies.
- ❖ The condition will be applied for each row on the table, if the condition returns true then that row is returned in the select query result.
- ❖ Text values are represented within single or double quotes
- ❖ Numeric values can be represented directly.
- ❖ Following are the list of comparison operators in MySQL:

Name	Description
>	Greater than operator
>=	Greater than or equal operator
<	Less than operator
<>, !=	Not equal operator
<=	Less than or equal operator
<=>	NULL-safe equal to operator
=	Equal operator
BETWEEN ... AND ...	Whether a value is within a range of values
IN()	Whether a value is within a set of values
IS	Test a value against a boolean
IS NOT	Test a value against a boolean
IS NOT NULL	NOT NULL value test
IS NULL	NULL value test
LIKE	Simple pattern matching
NOT BETWEEN ... AND ...	Whether a value is not within a range of values
NOT IN()	Whether a value is not within a set of values
NOT LIKE	Negation of simple pattern matching

- ❖ Logical operators help in combining multiple conditions:

Name	Description
AND, &&	Logical AND
NOT, !	Negates value
OR,	Logical OR
XOR	Logical XOR. Returns true if condition returns a different value

## ORDER BY

- ❖ The ORDER BY sort rows based on a column. It by default sorts in ascending order.
- ❖ The DESC clause helps to sort in descending order.

- ❖ Multiple columns can be provided separated by comma for ORDER BY. Wherever there is a repeating entry on the order by column, then the sorting will be done based on the second column provided in the order by.

#### Sub Query

- ❖ The concept of applying the result of one query to another is called as sub-query.
- ❖ In this example we have used '=' comparison operator against the sub-query, hence it is important the result of the sub-query returns only one row.
- ❖ If a sub-query returns multiple values then IN operator can be used.

#### Built-in Functions

- ❖ MySQL has numerous built in functions. The reference link below contains the full list.  
[https://www.w3schools.com/sql/sql\\_where.asp](https://www.w3schools.com/sql/sql_where.asp)

#### Check your understanding

- ❖ What does SQL stand for?
- ❖ SQL is case sensitive. [True / False]
- ❖ What is the purpose of WHERE clause?
- ❖ What is the syntax to apply a WHERE clause?
- ❖ List any three comparison operators.
- ❖ List the logical operators available in MySQL.
- ❖ What clause is used to sort the results of a select query?
- ❖ What clause is used to get the select query result in descending order?
- ❖ What is a sub-query?
- ❖ What is the syntax to write sub-query?

## Managing various sizes of the drink

### Problem

The coffee shop customers expect the drink to be available in various sizes, hence the shop decides to introduce two sizes small and large. Find below the pricing details for the same. Identify a way to store large prices in the menu table.

Menu Item	Small Price	Large Price
CAFFEE LATTE	\$ 2.70	\$ 3.25
CAPPUCCINO	\$ 2.70	\$ 3.25
CAFFEE MOCHA	\$ 3.00	\$ 3.55
ESPRESSO	\$ 1.55	\$ 1.85
ESPRESSO MACCHIATO	\$ 1.60	\$ 1.95

### Solution

Thought process on how to store the additional size related information in the database.

- ❖ How possibly can we include size related details in the menu table?
- ❖ Shall we add a new column large price to represent the new column?
  - One challenge in implementing this way is that it is not possible to map one menu item to one row. Each row will be representing two menu items.
  - Later, if one more size added then a new column needs to be added. The database design should ensure that we don't make structural modifications to tables quiet often.
  - There is also a possibility that certain items may not come in various sizes, which again complicates the design
- ❖ Another solution is to suffix each menu item name with SMALL and LARGE. We need to update the existing entries with SMALL and add new entries for LARGE.
  - The drawback here is that if we want to create sales report based on SMALL and LARGE items it makes it difficult.
- ❖ The final solution proposed is to add a new column for managing the size classification. Find below the list of changes to be done.
  - Modify the table to include a new column for SIZE.
  - Update the existing entries with SIZE column value as 'S' to represent small.
  - Insert new entries for large menu items with SIZE value as 'L' to represent large.
- ❖ Later if there is a new size coming up, it is sufficient to insert new rows and there is no need to modify the table structure by adding a new column.
- ❖ Execute the below statement to add a new column for size:

```
mysql> alter table menu add size char(1);
```
- ❖ Check the content of the menu table now. The size column has NULL values.



```
mysql> select * from menu;
```

item	price	size
CAFFEE LATTE	2.70	NULL
CAPPUCCINO	2.70	NULL
CAFFEE MOCHA	3.00	NULL
ESPRESSO	1.55	NULL
ESPRESSO MACCHIATO	1.60	NULL

```
5 rows in set (0.00 sec)
```

- ❖ Try running below query to demonstrate IS NULL and IS NOT NULL clause.

```
mysql> select * from menu where size is null;
```

item	price	size
CAFFEE LATTE	2.70	NULL
CAPPUCCINO	2.70	NULL
CAFFEE MOCHA	3.00	NULL
ESPRESSO	1.55	NULL
ESPRESSO MACCHIATO	1.60	NULL

```
5 rows in set (0.00 sec)
```

```
mysql> select * from menu where size is not null;
```

```
Empty set (0.00 sec)
```

- ❖ Execute the below update statement to change NULL values to 'S' to represent small size.

```
mysql> update menu set size = 'S';
```

- ❖ Execute select statement to check if the results are updated.
- ❖ Execute the following insert statements to add an items for large size.

```
mysql> insert into menu (item, price, size) values ('CAFFEE LATTEE', 3.25, 'L');
```

```
mysql> insert into menu (item, price, size) values ('CAPPUCCINO', 3.25, 'L');
```

```
mysql> insert into menu (item, price, size) values ('CAFFEE MOCHA', 3.55, 'L');
```

```
mysql> insert into menu (item, price, size) values ('ESPRESSO', 1.85, 'L');
```

```
mysql> insert into menu (item, price, size) values ('ESPRESSO MACCHIATO', 1.95, 'L');
```

- ❖ Run select query on menu and check if the data is stored as expected.
- ❖ Consider the scenario that large cappuccino price needs to be modified to a different value, then the update statement should look like this.

```
mysql> update menu set price = 2.95 where item = 'CAPPUCCINO' and size = 'L';
```

- ❖ Codd's 2<sup>nd</sup> rule is "Guaranteed Access Rule", which expects the possibility to identify a record based on a table name, primary key value and column name. This rule can be implemented using the primary key feature that can be applied to a table.
- ❖ In the above update statement, we are using more than one column to identify a row.
- ❖ If more columns get added to menu table, it becomes difficult to identify a row in menu table.
- ❖ Based on Codd's 2<sup>nd</sup> rule, it is recommended to add a primary key column.
- ❖ This column should be a numeric field and should get auto generated when a new record is inserted. Let us follow the steps below to get this incorporated.

- ❖ To create a new primary key column, let us execute the following command.

```
mysql> alter table menu add id int primary key;
```

- ❖ This should result in the following error:

```
ERROR 1062 (23000): Duplicate entry '0' for key 'menu.PRIMARY'
```

- ❖ The above statement adds the new column with the default value as 0. A primary key column should have unique values and it should not duplicate, since the value 0 is getting duplicated, the above command fails and the new column is not added.

- ❖ Execute the following command to add the new primary key column:

```
mysql> alter table menu add id int primary key auto_increment;
```

- ❖ Check if the primary key column is added and it has unique number assigned to each row.
- ❖ Now the way we write the insert statement due to auto increment primary key. Execute the below statement and check if a new record gets inserted.

```
mysql> insert into menu (item, size, price) values ('test', 'S', 2);
```

- ❖ Then run select command to check the result. A new entry is added with the incremented primary key value.
- ❖ Execute the delete command to remove the test entry using the where clause applied on the primary key column.

### Check your understanding

- ❖ What DDL command is used in order to add a new column to an existing table?
- ❖ What DML command is used to update an existing row in a table?
- ❖ What is the syntax for update command?
- ❖ Brief about primary key and its significance.
- ❖ What is the purpose of auto increment?
- ❖ How auto increment works when insert data into a table?

## Aggregating rows using Group By

### Problem

Problem A - Find the maximum prices under each size category. Find below the expected result.

size	max(price)
S	3.00
L	3.55

Problem B - Find the maximum prices under each size category that has the maximum price which is greater than 3 dollars. This should display only the L entry from the above result.

### Solution

- ❖ Execute the query below and check if expected result is obtained.

```
mysql> select size, max(price) from menu group by size;
```

- ❖ Execute the query below and check if expected result is obtained.

```
mysql> select size, max(price)
-> from menu
-> group by size
-> having max(price) > 3;
```

### Explanation

- ❖ GROUP BY clause helps to group rows based on one or more columns.
- ❖ In our example the GROUP BY is applied on the 'size' column. In this table there are two types of values for this column which is 'S' or 'L'.
- ❖ In the query result there are only two results, one having 'S' and the other having 'L'.
- ❖ In this group of records the maximum price is found and returned as result for max(price) column.
- ❖ Here max() is an built-in aggregate function.
- ❖ Execute the query below and check if expected result is obtained.
- ❖ The HAVING clause is similar to WHERE clause but it is applied on the aggregate result rather than each row.

### Check your understanding

- ❖ What is the purpose of GROUP BY clause?
- ❖ Explain the syntax on using GROUP BY clause.
- ❖ Explain the syntax on using GROUP BY and HAVING clause.

## Display Menu Items with Size

### Problem

The current menu table contains the size specification in separate columns, which makes it difficult to display as menu. Write a query or make modifications to table structure to achieve the result.

item	price
CAFFEE LATTE (Small)	2.70
CAPPUCCINO (Small)	2.70
CAFFEE MOCHA (Small)	3.00
ESPRESSO (Small)	1.55
ESPRESSO MACCHIATO (Small)	1.60
CAFFEE LATTEE (Large)	3.25
CAPPUCCINO (Large)	2.70
CAFFEE MOCHA (Large)	3.55
ESPRESSO (Large)	1.85
ESPRESSO MACCHIATO (Large)	1.95

### Solution (using case clause)

- ❖ Execute the query below and check if expected result is obtained.

```
mysql> select item,  
-> case  
->   when size = 'S' then " (Small)"  
->   when size = 'L' then " (Large)"  
-> end size,  
-> price  
-> from menu;
```

- ❖ Modify the query to display the item and name in a single column.

```
mysql> select  
-> case  
->   when size = 'S' then concat(item, " (Small)")  
->   when size = 'L' then concat(item, " (Large)")  
-> end item,  
-> price  
-> from menu;
```

### Solution (using normalization)

- ❖ Consider the scenario where a new size for medium is introduced.
- ❖ In this scenario the above query needs to be modified to get the result, which requires development and testing effort.
- ❖ Now, we will apply the concept related to normalization that avoids redundant data.
- ❖ The information related to Small or Large can be moved to a separate table.
- ❖ Follow steps below to incorporate this change.
- ❖ Create a new table to store the size details.

```
mysql> create table drink_size (
-> id int primary key auto_increment,
-> name varchar(20)
-> );
```

- ❖ Insert necessary data into the new table.

```
mysql> insert into drink_size (name) values ('Small');
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into drink_size (name) values ('Large');
Query OK, 1 row affected (0.00 sec)
```

- ❖ Modify the menu table to include a new column for linking the new table.

```
mysql> alter table menu add column size_id int;
```

- ❖ Execute the below commands to modify the new column with id value that points the respective size in the drink\_size table.

```
mysql> update menu set size_id = 1 where size = 'S';
```

```
mysql> update menu set size_id = 2 where size = 'L';
```

- ❖ Remove the size column, which is not required any more.

```
mysql> alter table menu drop column size;
```

- ❖ The following query gets the data combining both the tables.

```
mysql> select item, name, price from menu, drink_size where size_id = id;
ERROR 1052 (23000): Column 'id' in where clause is ambiguous
```

- ❖ Both menu and drink\_size table contains a column named 'id' columns. During the execution of the query, MySQL is unable to identify which id column to use. Use the query below which provides reference to which column the table belongs to.

```
mysql> select item, name, price
-> from menu, drink_size
-> where size_id = drink_size.id;
```

- ❖ Modify the above query with alias names for tables.

```
mysql> select item, name, price
-> from menu m, drink_size s
-> where size_id = s.id;
```

## Explanation

- ❖ The alter table drop column clause helps in removing a column from a table.
- ❖ The query part 'menu m' specifies an alias name for 'menu' table as m.
- ❖ Similarly, 's' is defined as the alias name for 'drink\_size' table.
- ❖ The query executed is called as inner join.
- ❖ To understand joins let us execute the below query.

```
mysql> select * from menu m, drink_size s;
```

- ❖ This query will result in a combination of all rows between menu table and drink\_size table.
- ❖ The number of rows returned will be a multiplication of number of rows between the two tables.
- ❖ The 'menu' table contains 10 rows. The 'drink\_size' table contains 2 rows. The total rows returned is 20 rows which multiplication of 10 and 2.

- ❖ This kind of query is called Cross Join or Cartesian Product.
- ❖ In the query result obtained compare values of 'size\_id' column in 'menu' table and 'id' column in 'drink\_size' table. This comparison condition is what is done in the inner join query.

```
mysql> select item, name, price
-> from menu m, drink_size s
-> where size_id = s.id;
```

- ❖ This kind of join query definition is called as 'implicit join notation'.

## Summary

- ❖ Cartesian Product or Cross Join provides result which contains all combinations of rows in all the tables.
- ❖ Inner Join filters out the Cartesian Product result using a join condition.
- ❖ Creating a join query without explicitly mentioning the 'join' keyword is called as 'implicit join notation'.

## Solution (concatenating item name and size)

- ❖ To put the item name and size together, the following query can be used.

```
mysql> select concat(item, '(', name, ')'), price
-> from menu m, drink_size s
-> where size_id = s.id;
```

- ❖ The item column name is not as expected which can be changed by providing the query below.

```
mysql> select concat(item, '(', name, ')') item, price
-> from menu m, drink_size s
-> where size_id = s.id;
```

## Explanation

- ❖ 'concat()' is a built-in method on MySQL that concatenates the input parameters provided.
- ❖ In the last query executed, an alias name had been provided to the concat() function definition.

## Solution (converting the join with explicit join notation)

- ❖ Execute below query that runs with explicit join notation.

```
mysql> select item, name, price from menu m join drink_size s on size_id = s.id;
```

- ❖ Modify the above query with concatenation of item and size

```
mysql> select concat(item, '(', name, ')') item, price
-> from menu m join drink_size s on size_id = s.id;
```

## Explanation

- ❖ Notice the explicit definition of 'join' between the tables
- ❖ The 'on' clause helps in defining the join condition

- ❖ The query can also be changed with inclusion of 'inner' clause before the join, which gives the same result.

```
mysql> select concat(item, ' (', name, ')') item, price  
-> from menu m inner join drink_size s on size_id = s.id;
```

### Solution (defining foreign key)

- ❖ One of the draw backs in the previous implementation is that the size\_id column in menu table allows values even if it is not present in the drink\_size table.
- ❖ Try executing the following query:

```
mysql> update menu set size_id = 3 where id = 10;
```

- ❖ Now the menu item with id value as 10 points to a drink\_size which is non-existent.
- ❖ To avoid having this sort data integrity issues, it is recommended to define size\_id column as a foreign key column referring to drink\_size table's id column.
- ❖ Execute the following query to alter the menu table to be included with the foreign key constraint.

```
mysql> alter table menu add constraint fk_size foreign key (size_id)  
-> references drink_size(id);
```

- ❖ The above query fails because of the reason that item 10 is having value as 3 which is non-existent, execute the below query to revert back the data to the previous state.

```
mysql> update menu set size_id = 2 where id = 10;
```

- ❖ Execute the alter table query again to include the foreign key constraint.

```
mysql> alter table menu add constraint fk_size foreign key (size_id)  
-> references drink_size(id);
```

- ❖ Now try running the below update command again which will now fail because of the foreign key constraint.

```
mysql> update menu set size_id = 3 where id = 10;
```

### Explanation

- ❖ The 'add constraint' clause helps in adding a new constraint for the menu table.
- ❖ 'fk\_size' refers to the name provided to the foreign key constraint.
- ❖ The 'foreign key' clause helps in defining a foreign key constraint.
- ❖ The '(size)' refers to the column in the current table which needs to be linked to the other table.
- ❖ The 'references' clause helps in creating the link to the other table
- ❖ The 'drink\_size(id)' helps in defining the other table and column where the foreign key needs to be applied.
- ❖ The foreign key constraint definition helps to ensure that the integrity between the data is maintained.

### Check your understanding

- ❖ What is the syntax to remove a column from a table?

- ❖ What is the advantage of defining alias names for tables in a select query?
- ❖ What is the syntax to define the alias names for tables?
- ❖ What is a Cartesian Product or Cross Join?
- ❖ What is inner join?
- ❖ What is the syntax to define an Implicit Join Notation?
- ❖ What is the syntax to define an Explicit Join Notation?
- ❖ What is the significance of a foreign key constraint?
- ❖ What is the syntax to define a foreign key constraint?



## Find absentees using outer join

### Problem

There is a student table and an attendance table available in a database. List the student attendance details of a specific day. The query result should contain the absentees as well.

Execute below scripts to create necessary tables for student and attendance. The attendance needs to be listed for 15 Jan 2020.

```
create table student(  
  st_id int primary key auto_increment,  
  st_name varchar(30)  
);  
insert into student (st_name) values ('John');  
insert into student (st_name) values ('Jack');  
insert into student (st_name) values ('James');  
insert into student (st_name) values ('Stella');  
  
create table attendance(  
  at_id int primary key auto_increment,  
  at_st_id int,  
  at_date date,  
  constraint at_st_fk foreign key (at_st_id) references student(st_id)  
);  
insert into attendance (at_st_id, at_date) values (1, '2020-01-15');  
insert into attendance (at_st_id, at_date) values (2, '2020-01-15');  
insert into attendance (at_st_id, at_date) values (4, '2020-01-15');
```

### Solution

- ❖ Understanding the existing data.
  - There are four students in student table.
  - The attendance table contains the student id and the attendance date.
  - An entry in the attendance table means that the respective student was present on that date.
  - The entry of an absentee will not be present in the attendance table.
  - For the date Jan 1, 2015 James was absent, hence his entry is not present in the attendance table.

- ❖ Let us try to solve the problem using inner join query below.

```
mysql> select at_date, st_name  
-> from student  
-> inner join attendance on st_id = at_st_id  
-> where at_date = '2020-01-15';
```

- ❖ Notice that the result does not contain the absentee James. Outer join helps to bring out unmatched entries as well. Execute the query below.

```
mysql> select at_date, st_name  
-> from student  
-> left outer join attendance on st_id = at_st_id;
```

- ❖ Now we will try to include date condition

```
mysql> select at_date, st_name  
-> from student  
-> left outer join attendance on st_id = at_st_id  
-> where at_date = '2020-01-15';
```

- ❖ Now the result is not returned since at\_date is returned as NULL, which is not part of the filter criteria, hence we will include the respective condition.

```
mysql> select at_date, st_name  
-> from student  
-> left outer join attendance on st_id = at_st_id  
-> where at_date = '2020-01-15' or at_date is null;
```

- ❖ The absentee entry is displayed as NULL, using query below changes the value to 'ABSENT'.

```
mysql> select ifnull(at_date, 'ABSENT') date, st_name  
-> from student  
-> left outer join attendance on st_id = at_st_id  
-> where at_date = '2020-01-15' or at_date is null;
```

### Explanation

- ❖ Value for date is defined in 'yyyy-mm-dd' format.
- ❖ The clause 'left outer' helps to bring non-matching data from the left-hand side table.
- ❖ In this query, the table in the left-hand side is student and the table in right hand side is attendance.
- ❖ If the position of table definitions is interchanged then the clause needs to be changed as right outer join.

### Explanation for table and column naming convention

- ❖ Identify a two-character unique alias for each table in a database
- ❖ Prefix the column name with the alias followed by underscore.
- ❖ For example, for 'student' table the alias is identified as 'st', hence 'st' is prefixed before each column name. Similarly, for 'attendance' table, 'at' is identified as alias name.
- ❖ If there is a foreign key column include the foreign key tables alias name. For example, student id column is present in the attendance table, which is represented as at\_st\_id.
- ❖ This is one of the naming convention followed, which helps in uniquely identifying any column in a database, this avoids the need to specify alias name when writing join queries.

### Check your understanding

- ❖ What is outer join and what is its purpose?
- ❖ Explain the syntax to define an outer join.

## Create stored procedure to insert data

### Problem

Create stored procedure to insert data into drink\_size table.

### Solution

- ❖ Open MySQL Workbench
- ❖ Connect to the local database.
- ❖ Double click on 'learnadb' which is equivalent to the 'use' command.
- ❖ Select File > New Query Tab
- ❖ Copy and paste the below script in the new window, which creates a new stored procedure in the database.

```
DELIMITER //  
Create PROCEDURE insert_size(IN p_name varchar(20))  
BEGIN  
    insert into drink_size(name) values (p_name);  
END //  
DELIMITER ;
```

- ❖ Execute the above statements by using the lightning icon in Workbench.
- ❖ Click on the Refresh icon next to the SCHEMAS heading in the "Schemas" window in the left-hand side.
- ❖ Expand "Store Procedures" and we can see the new stored procedure listed.
- ❖ Open a new editor window using File > New Query Tab and run the below command.
- ❖ Execute the stored procedure to check if it is working fine.

```
call insert_size('Medium');
```

- ❖ Expand Tables in the Schema section. Right click on 'drink\_size' table and select "Select Rows", which will display the result in a new window.
- ❖ Check the result if it contains the new entry for Medium size.
- ❖ It is also possible to run the call command from mysql client program.

### Explanation

#### Syntax

- ❖ The DELIMITER command changes the statement delimited from ; to //
- ❖ If the delimiter is not changed, then the statements of the stored procedure will be executed line by line.
- ❖ Currently we do not want to execute the statements in the stored procedure line by line, instead we want to compile the stored procedure as a whole, hence a // is provided after the END statement which denotes the end of the stored procedure.
- ❖ After END// we are reverting back the delimited to ; using the DELIMITER command again.

- ❖ Create PROCEDURE is the syntax used to create a new stored procedure, which is followed by the procedure name, after which the parameters of the stored procedure is defined with respective data types.
- ❖ IN represents the input parameter.
- ❖ There are three types of parameters:
  - IN – Parameter value that needs to be passed by the caller
  - OUT – These types of parameters can be used to return back value to the calling program
  - INOUT – These types of parameters are used to providing input to the stored procedure and can be used to respond back values.
- ❖ BEGIN starts the body of the stored procedure.
- ❖ END ends the body of the stored procedure.
- ❖ When the stored procedure is invoked using the call command, the lines between the BEGIN and END is executed.
- ❖ The p\_name passed as input is included in the insert statement which is what will be used to insert into the drink\_size table.

#### About Stored Procedures

- ❖ A stored procedure is a set of SQL statements stored inside the MySQL database server
- ❖ In this example we have created a stored procedure named 'insert\_size'
- ❖ When the create procedure script was executed, the stored procedure got compiled and stored in the database.
- ❖ Once created a stored procedure can be executed multiple times, the compilation step happens only once and calls made the stored procedure does not compile it again.
- ❖ The 'call' command is used to execute a stored procedure.
- ❖ A stored procedure can have input and output parameters.
- ❖ Input parameters are used to send value to the stored procedure.
- ❖ Output parameters are used to receive value back from the stored procedure.
- ❖ Use DROP PROCEDURE [procedure\_name] to remove an existing procedure.
- ❖ It is not possible to modify a Stored Procedure, it needs to be dropped and created again, inclusion of below line after the first DELIMITER definition ensures that the script can be used again and again to modify a function.

```
drop procedure if exists insert_size;
```

- ❖ A stored procedure can
  - execute SQL statements
  - define local variables
  - contain control flow statements such as IF and CASE.
  - contain looping constructs.
  - call other stored procedures or stored functions.

#### Advantages

- ❖ Reduce network traffic
  - Stored Procedures are typically called from a client application, which is available in a network

- If there are multiple SQL operations that needs to be performed, then each SQL statement has to go through the network from the client.
- Instead if we use stored procedure, all the SQL operations can be mentioned in a stored procedure, this helps in reducing the number of calls made.
- ❖ Business Logic
  - The application logic can all be defined in stored procedures.
  - This makes all business logic to be centralized and can be reused by multiple clients.

#### Disadvantages

- ❖ Having many stored procedures increases memory usage
- ❖ Large number of logical operations in stored procedure increases the CPU usage
- ❖ It is difficult to debug a stored procedure.
- ❖ MySQL stored procedure does not have any debugging tools available
- ❖ Maintaining a large set of stored procedures requires specialized skills

#### Check your understanding

- ❖ What is a stored procedure?
- ❖ What are the advantages of using a stored procedure?
- ❖ What are the disadvantages of using a stored procedure?
- ❖ What is the purpose of the DELIMITER command?
- ❖ What is the syntax for creating a stored procedure?
- ❖ What are the three types of parameter that can be defined in a stored procedure? Brief about how each one of it works.
- ❖ What kind of statements can be defined within a stored procedure?

#### Additional information related to Stored Procedures

- ❖ Declaring variable and assign a value.
 

```
declare total int;
set total = 10;
```
- ❖ Assign query result into variable.
 

```
select count(menu) into total from menu;
```
- ❖ IF ELSE statement. For multiple conditions ELSEIF can be used
 

```
if balance < 5000 then
  set pMessage = "Insufficient Balance";
else
  set balance = balance - transaction_amount;
end if;
```
- ❖ Example for CASE statement

```

case country_code
  when 'US' then
    set shipping_charge = 5;
  when 'IN' then
    set shipping_charge = 2;
  else
    set shipping_charge = 10;
end case;

```

- ❖ Looping based on label

```

loop_label: loop
  if count = 10 then
    leave loop_label;
  end if;
  count = count + 1;
end loop;

```

- ❖ WHILE loop example

```

while count = 10 do
  count = count + 1;
end while;

```

- ❖ REPEAT loop example. This loop gets executed at least one time.

```

repeat
  count = count + 1;
until count = 10;

```

- ❖ Exceptions that happen during the execution of a program can be handled using exception handling options available. (Refer: <https://www.mysqltutorial.org/mysql-error-handling-in-stored-procedures/>)
- ❖ To create an exception SIGNAL and RESIGNAL can be used. (Refer: <https://www.mysqltutorial.org/mysql-signal-resignal/>)
- ❖ CURSOR can be used to perform row by row processing on a select query result. (Refer: <https://www.mysqltutorial.org/mysql-cursor/>)

## Create stored function to display ABSENT instead of null

### Problem

The following query was used to display attendance details with display as 'ABSENT' when the value returned is null.

```
select ifnull(at_date, 'ABSENT') date, st_name
from student
left outer join attendance on st_id = at_st_id
where at_date = '2020-01-15' or at_date is null;
```

In this query use a stored function instead of using ifnull() function.

### Solution

- ❖ Open MySQL Workbench
- ❖ Connect to the local database.
- ❖ Double click on 'learnadb' which is equivalent to the 'use' command.
- ❖ Select File > New Query Tab
- ❖ Copy and paste the below script in the new window, which creates a new stored function in the database.

```
delimiter //
create function absent(timesheet_date date) returns varchar(30)
no sql
begin
    if timesheet_date is null then
        return 'ABSENT';
    else
        return timesheet_date;
    end if;
end//
delimiter ;
```

- ❖ Execute the above statements by using the lightning icon in Workbench.
- ❖ Click on the Refresh icon next to the SCHEMAS heading in the "Schemas" window in the left-hand side.
- ❖ Expand "Store Functions" and we can see the new stored function 'absent' listed.
- ❖ Open mysql client command line window and execute the below query.

```
mysql> select absent(at_date) date, st_name
-> from student
-> left outer join attendance on st_id = at_st_id
-> where at_date = '2020-01-15' or at_date is null;
```

- ❖ The above query should return the same result with 'ABSENT' listed.

## Explanation

### Syntax

- ❖ The CREATE FUNCTION helps in creating a function followed by the name of the function
- ❖ Similar to procedure parameters can be passed as input
- ❖ A function must return a value hence we need to provide a datatype with return keyword
- ❖ The keywords "no sql" helps to define that the function does not contain any SQL operations
- ❖ Within the body of the function we have defined if condition to check if the date is null or not and based on that we are responding with a result.
- ❖ The 'return' statement helps to return the value to the caller of the function.
- ❖ Since a function has to return a value, it can be directly applied in a column.

### About Stored Function

- ❖ A special kind of store program that returns a single value
- ❖ Used for encapsulating common formulas or business rule
- ❖ A Stored Function can be reused in Stored Procedures, Stored Functions and select queries
- ❖ Use Stored Function wherever an expression is used, which improves the readability and maintainability of the code.
- ❖ Use DROP FUNCTION [function\_name] to remove a function from the database.
- ❖ It is not possible to modify a function, it needs to be dropped and remove, inclusion of below after the first DELIMITER definition ensures that the script can be used again and again to modify a function.

```
drop function if exists absent;
```

## Check your understanding

- ❖ What is a stored function?
- ❖ What are the advantages of using a stored function?
- ❖ What is the syntax for creating a stored function?
- ❖ How does stored function differs from a stored procedure? Mention syntax level differences as well.



## Record all INSERT, UPDATE and DELETE operations on a table using triggers

### Problem

Consider the following scenarios:

- ❖ A menu item is updated with a new price in the 'menu' table. Is it possible to know the price that was there earlier?
- ❖ A menu item is deleted from the 'menu' table. At a later point of time, we want to know about this menu item that had been deleted and it's earlier price. How can we know about this?

Arrive a solution to handle the above scenarios.

### Solution

- ❖ This problem can be solved using triggers.
- ❖ A new audit table is going to store all the changes that are going to happen in the menu table.
- ❖ This table will contain all the columns of the menu table with an additional id column to identify each row.
- ❖ Execute the below script to create the audit table.

```
create table menu_audit(  
    id int primary key auto_increment,  
    menu_id int,  
    action varchar(6),  
    item varchar(30),  
    price decimal(5,2),  
    size_id int,  
    change_time datetime  
);
```

- ❖ Execute the below script to create an insert trigger.

```
create trigger menu_insert_audit_trigger  
after insert on menu  
for each row  
insert into menu_audit (  
    menu_id, action, item, price, size_id, change_time  
) values (  
    NEW.id, 'INSERT', NEW.item, NEW.price, NEW.size_id, now()  
);
```

- ❖ Execute the below script to create an update trigger.

```
create trigger menu_update_audit_trigger  
before update on menu  
for each row  
insert into menu_audit (  
    menu_id, action, item, price, size_id, change_time  
) values (  
    NEW.id, 'UPDATE', NEW.item, NEW.price, NEW.size_id, now()  
);
```

- ❖ Execute the below script to create a delete trigger.

```
create trigger menu_delete_audit_trigger  
before delete on menu  
for each row  
insert into menu_audit (  
    menu_id, action, item, price, size_id, change_time  
) values (  
    OLD.id, 'DELETE', OLD.item, OLD.price, OLD.size_id, now()  
);
```

- ❖ Execute the below insert command in 'menu' table.

```
insert into menu (item, price, size_id) values ('CAFFEE AMERICANO', 2.5, 1);
```

- ❖ Execute the below update command in 'menu' table.

```
update menu set item = 'CAFFEE AMERICAN', price = 3.5, size_id = 2  
where item = 'CAFFEE AMERICANO';
```

- ❖ Execute the below delete command in 'menu' table.

```
delete from menu where item = 'CAFFEE AMERICAN';
```

- ❖ Now run a select query on the audit table, it should contain the details about all the insert update and delete we did just now.

```
select * from menu_audit;
```

## Explanation

### About Triggers

- ❖ Triggers are small stored programs that are executed automatically when insert or update or delete operation is performed on a table.
- ❖ Triggers help in tracking the changes in a table. This is also called as an audit.
- ❖ Triggers also help in maintaining the integrity of the data by altering data to a valid data before and insert or update.

### Syntax

- ❖ The 'CREATE TRIGGER' command helps to create a new trigger.
- ❖ The 'BEFORE' or 'AFTER' clause helps in defining whether the trigger needs to be executed before execution of the command or after execution of the command.
- ❖ The 'INSERT/UPDATE/DELETE' clause helps in specifying the operation during which the trigger needs to be executed.
- ❖ The table on which the trigger needs to be executed using the 'ON' clause.
- ❖ The clause 'FOR EACH ROW' specifies that the trigger needs to be executed for each row that is inserted, updated or deleted. Consider that when an update statement is executed, it is possible that it may update multiple rows, in this scenario we define that the trigger needs to be executed for each row.
- ❖ If a trigger has multiple statements of execution, then the execution lines should be enclosed within a BEGIN and END.
- ❖ Since there is only one line for execution, we have only one insert statement in the body of the trigger, hence it is not enclosed within a BEGIN and END.
- ❖ The clause NEW and OLD is followed by the column name. This helps to get the old value before execution of the statement and the new value that is defined as part of the statement.
- ❖ INSERT statements will only have NEW value.
- ❖ UPDATE statements will have both OLD value and NEW value.
- ❖ DELETE statements will only have OLD value.

### Check your understanding

- ❖ What is a trigger?
- ❖ In what scenarios can we use trigger?
- ❖ What is the significance of BEFORE and AFTER clause?
- ❖ What does the NEW and OLD represent within a trigger body?

## Design database for coffee shop

### Problem

Design a database for coffee shop chain.

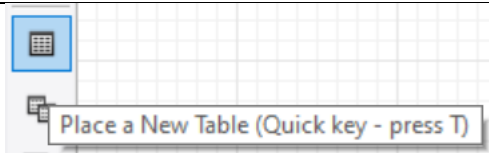
- ❖ It has 30 branches
- ❖ The menu and price are same in each branch
- ❖ Manage the menu items
- ❖ Based on the menu items it should be possible to create a receipt. Each receipt should be available in the database, so that it can be retrieved at any later point time.

Coffee Shop Main Street Branch			
18 Jun 2020			
-----			
Item	Qty	Price	Total
-----			
ESPR. MACCH. (Large)	2	1.95	3.90
CAFF. MOCHA (Small)	1	3.00	3.00
CAFF. LATTEE (Small)	3	2.70	8.10
-----			
TOTAL			15.00
-----			
THANK YOU!!!			

- ❖ The database design should be created as an **ER Diagram**
- ❖ It is sufficient to design and ER Diagram, user interface is not required.

### Solution

- ❖ The first step to identify the list of entities that needs to be considered for the coffee shop.
- ❖ Then we need to identify the relationships between the entities. The relationship can fall under any one of the below categories:
  - One to One
  - One to Many
  - Many to One
  - Many to Many
- ❖ Then identify the attributes for each entity.
- ❖ Open MySQL Workbench
- ❖ Select File > New Model
- ❖ Double click on "Add Diagram"
- ❖ Steps to create a new table and add columns
  - Click the below specified icon and click within the create a new table.



- Double click on the new table, which will open a new section in the bottom
- Modify the table name in this section

table1 - Table x

Table Name:

- Click under the "Column Name", which will create a text field for entering the column name.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idtable1	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- Enter the primary key column name and check the box under AI. Refer sample below.

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
st_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

- Click below column name to add new columns. Specify the relevant data type.

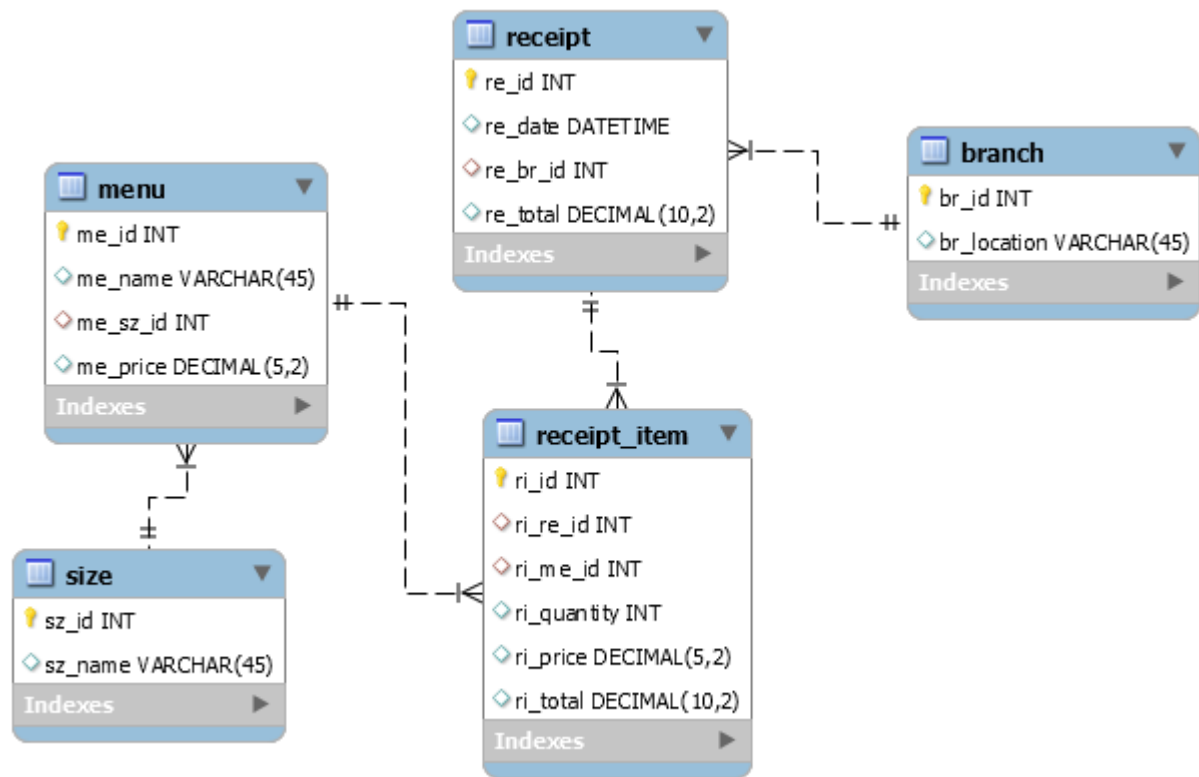
❖ Steps to define foreign key:

- Double click on the table and click "Foreign Keys" tab in the bottom
- Enter the "Foreign Key Name" with 'fk' as suffix.
- In "Referenced Table" select the table that this column has to refer
- In the right-hand side select the foreign key column and the referencing column.

Foreign Key Name	Referenced Table
me_sz_fk	'coffeeshop', 'size'

Column	Referenced Column
<input type="checkbox"/> me_id	
<input type="checkbox"/> me_name	
<input checked="" type="checkbox"/> me_sz_id	sz_id
<input type="checkbox"/> me_price	

❖ Refer the diagram below.



## Explanation

- ❖ An ER Diagram has three main components:
  - Entity
  - Relationship
  - Attribute
- ❖ **Entity** – An entity represents a real-world object. Refer examples below. This can be related to a database table.
  - City
  - Country
  - Passport
  - Employee
  - Ticket
  - Bank Transaction
  - Bank Statement
  - Savings Account
- ❖ **Relationship** – Defines how entities are related to each other. This can be related to foreign key constraints in a table.
  - One to One
    - One Person has one Passport
    - One Person lives in one Address
  - One to Many

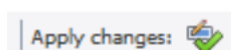
- One Department has many Employees
- One Train has many Coaches
- One Coffee Shop Branch has many Receipts
- Many to One
  - Many Employees belong to One Department
  - Many Coaches belong to One Train
  - Many Receipts can be generated by One Coffee Shop Branch
- Many to Many
  - One Employee can have many Skills  
One Skill can be acquired by many Employees
  - One Book can have many Authors  
One Author can write many Books
  - One Movie can be screen in many Screens  
One Screen can show many Movies
  - One Receipt can have many Menu Items  
One Menu Item can be part of many Receipts
- ❖ **Attributes** – Attributes represents the property or characteristics of an entity. This can be related to the columns in a table.
  - City – Name, Population, Area
  - Passport – Number, First Name, Last Name, Date of Birth
  - Ticket – Travel Date, From, To, Price

### Create database based on the ER diagram

- ❖ Ensure the column order is defined in the same order as specified in the above diagram.
- ❖ Find below the steps to add data into tables:
  - Double click on 'size' table
  - Click "Inserts" tab
  - Enter the data as specified below:

	sz_id	sz_name
	1	Small
	2	Large
	NULL	NULL



- Then click apply changes to data icon to save the details:



- Add these entries in branch table:

	br_id	br_location
	1	Main Street
	2	High Street
	NULL	NULL

- Using the table below key in the menu details.

- ❖ Create a new folder named 'mysql-learn' in a convenient folder, so that it can be referred in the future.
- ❖ Download below CSV files into mysql-learn folder. These files contain the data for menu, receipt and receipt\_item tables
  - menu - [https://drive.google.com/file/d/1TogNtc7yK\\_fUXhdbJ\\_zDFB7Fj-jAwYwO/view?usp=sharing](https://drive.google.com/file/d/1TogNtc7yK_fUXhdbJ_zDFB7Fj-jAwYwO/view?usp=sharing)
  - receipt - [https://drive.google.com/file/d/18gxyrdt-B-8ekHBlrXSJgiKLV7\\_xodqy/view?usp=sharing](https://drive.google.com/file/d/18gxyrdt-B-8ekHBlrXSJgiKLV7_xodqy/view?usp=sharing)
  - receipt-item - <https://drive.google.com/file/d/14sAQInVMt7cM49q3XFeP472nhfSEhftT/view?usp=sharing>
- ❖ Follow steps below to import the data into respective tables.
  - Double click on menu table in workbench ER diagram
  - Select 'Inserts' tab
  - Click on "Import records from external file" option
 
  - Select the respective downloaded file, which will populate the data into the table.
  - Click on "Apply changes to data" icon
 
  - Repeat the above steps for receipt and receipt-items tables
- ❖ Save the model file using File > Save Model option. Name the file as coffeeshop and save the file in mysql-learn folder.
- ❖ Follow steps below to create the database based on the ER Diagram.
  - In Workbench, go to File > Export > "Forward Engineer SQL CREATE Script.."
  - Click on the button on the top right corner and select the mysql-learn folder and give the file name as "coffeeshop"
  - In the window that appears check the following options:
    - Generate Drop Schema
    - Generate insert statements for tables
  - Click "Next"
  - Click "Next" again
  - Scroll through the SQL script and check on the following aspects.
    - Schema Creation
    - Table Creation
    - Insert Statements
  - Click "Finish"
  - Check if a file named coffeeshop.sql is created in the mysql-learn folder.
  - Open mysql client in command prompt.
  - Run the below command in the mysql command line.

```
source [replace-here-with-mysql-learn-folder]\coffeeshop.sql
```



- This should create the database, tables and insert necessary data
- If there is script failure due to some special characters in the first line of the insert, open coffeeshop.sql file in Notepad++ and press backspace just before the id 1 of each table. This will not remove the bracket, which means that the special character is removed.
- Save the file after the above changes and run the source command again to check if the SQL script runs successfully or not.
- Run a select query on the tables to check if data is present.
- Execute the following queries.
- Query to get all the receipts on a specific date for both the branches.

```
select re_id receipt_id,
       date_format(re_date, '%d %b %y') date,
       br_location branch,
       re_total receipt_total
from   receipt
join   branch on re_br_id = br_id
where  re_date = '2020-01-08';
```

- Query to get the receipt details of based on a specific receipt id.

```
select date_format(re_date, '%d %b %y') date,
       br_location,
       concat(me_name, ' (', sz_name, ')') item,
       ri_quantity,
       ri_price,
       ri_total,
       re_total
from   receipt
join   receipt_item on re_id = ri_re_id
join   menu         on me_id = ri_me_id
join   branch       on br_id = re_br_id
join   size         on sz_id = me_sz_id
where  re_id = 1;
```

- Query to get the day wise branch sales

```
select br_location,
       date_format(re_date, '%d %b %y') date,
       sum(re_total)
from   receipt
join   branch on br_id = re_br_id
group by br_id, re_date;
```

## Exercise Problems

- ❖ For the below provided topics identify the entity. It is sufficient to list the entities
  - Online food aggregation app
  - Online shopping
  - School
  - Library
  - Training Institute
- ❖ For each of the above identified entities, define the relationship type between the entities. Draw it in a paper and send the image through WhatsApp.

## Miscellaneous Topics

### DISTINCT

- ❖ Use DISTINCT clause after SELECT clause to remove duplicates in the select query result.

### SELF-JOIN

- ❖ A table can be joined to itself, which is called as self-join.
- ❖ Example
  - An employee table can contain a manager\_id column representing the manager of an employee.
  - This manager\_id points to another employee in the employee table itself.
  - The employee id of the manager will be stored in the manager\_id column.
  - To get the manager name of each employee a self-join needs to be made to employee table itself.

### UNION

- ❖ Use UNION to join the result of two queries having same set of columns:

```
(select query) UNION (select query)
```

### INTERSECT

- ❖ Use intersect when we need to get matching entries from two tables.
- ❖ In MySQL there is no INTERSECT clause available and this can be achieved using subquery and IN operator.

### INSERT into SELECT

- ❖ Use this clause when we need to insert a select query result into another table.

### COMMIT and ROLLBACK

- ❖ Run the following command in mysql client to disable auto committing.

```
set autocommit = 0;
```

- ❖ Insert a record into one table.
- ❖ Display the records in that table using select query, it should display the newly inserted record.
- ❖ Open another mysql client window and check if the new record inserted can be seen. It should not be visible.
- ❖ Execute the 'COMMIT;' command in the first mysql client window and then check the table data in the second mysql client window. Now it should display the data.
- ❖ Insert a record into one table.
- ❖ Display the records in that table using select query, it should display the newly inserted record.
- ❖ Run the ROLLBACK; command and run the select query. The newly added record should be gone now.

### CONSTRAINTS

- ❖ Apart from Primary Key and Foreign Key constraints there are other constraints available. Let us understand them one by one:
  - NOT NULL constraint – NULL value cannot be inserted into a column
  - UNIQUE constraint – Does not allow duplicate values in a column. An username column is an ideal candidate for this constraint, it ensures that username is not duplicated.

- CHECK constraint – Evaluates the value of a column. Refer examples below:

```
age int not null check (age > 0)
```

## LOAD DATA INFILE

- ❖ Use the above command to load data from a CSV (Comma Separated Values) file into a table.

## SELECT INTO OUTFILE

- ❖ Using 'SELECT INTO OUTFILE' clause it is possible to create a CSV file containing the select query result.

## mysqldump

- ❖ The 'mysqldump' command can be used like 'mysql' and 'mysqld' commands to migrate an entire database into a SQL file.
- ❖ The SQL file can be executed in another database to create the same tables and data.

## Views

- ❖ A complex select query can be converted into a view to improve the readability. The syntax for creating a view is specified below:

```
create view as [complex select query]
```

## Indexes

- ❖ As our database starts growing larger, the results of select query are going to take longer time.
- ❖ It is always a good idea to create indexes for columns we use in the where clause.
- ❖ Indexes create a separate structure to quickly identify rows in table.
- ❖ Following is the syntax to create an index:

```
create index index_name on table_name (column_list)
```