# Objectives

INFO90002 Tutorial Week 5

* Learn how to type SQL SELECT commands
* Learn how to FILTER by row and column.
* Learn to use maths conditions
* Learn how to use ORDER BY, LIMIT
* Learn how to use FUNCTIONS (max, min, sum, avg)
* Learn how to use GROUP BY
* Learn hot to add your student ID to the result set

### Using MySQL Workbench, connect to your database to the MySQL Server

1. Download the labs2018engv6.sql script from the LMS (labs2018byodv6.sql if on your own device)

### From the SQL Query window Choose File -> Open SQL Script

Select the place where you downloaded the labs2018v6.sql script

### Click the lightening icon to execute

1. Show the catalogue (meta data) about the Department table

### DESC Department;

Hint: in labs, if you get an error message or can’t find the Department table, you may also need to tell the

database to use the correct schema

### use <username>;

FOR BYOD machines

use labs2018;

# SELECT

SELECT statements retrieve and display data from the database. The structure of a SELECT statement is below:

**SELECT** (select list)

**FROM** (from list)

### **WHERE** (filtering list) -- optional

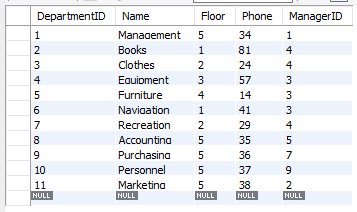
**GROUP BY** (grouping list) -- optional **HAVING** (group qualifications) –- optional **ORDER BY** (grouping list) –- optional

### **LIMIT** (number of rows) OFFSET (number of rows) -- optional

To select all columns from a table we use the SQL shorthand "\*" To select from the Department table, enter the following SQL:

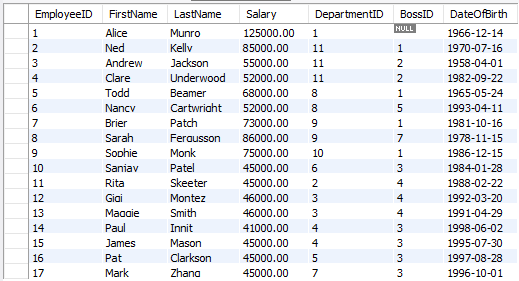
**SELECT** \*

**FROM** department;



### Type the SQL query to select all rows and columns from the Employee table.

You should see a result like this:



# Filtering

To select only some columns, we specify the columns we want in the query. Separate each column name with

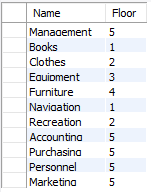
a “,”. If you describe the Department table we can see attributes Name and Floor, amongst others.

### DESC department;

To select only these two columns in the Department table, enter this SQL:

**SELECT** Name, Floor

**FROM** department;

You should see a result set like this:

### A screenshot of a cell phone Description automatically generatedType the SQL query to select the first name, last name and departmentid in the Employee table.

Until now we have selected **all** the rows in a table. Most times we don’t want to retrieve all rows.

In SQL we do this by using a *selection condition* on our query. This is also known as a *selection* in Relational Algebra. If for example, we wished to list all Departments that are listed on the second floor:

**SELECT** \*

**FROM** department

**WHERE** floor = 2;

A screenshot of a cell phone  Description automatically generated

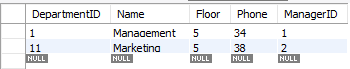
The SQL keyword ‘WHERE’ lists any horizontal filters we wish to specify for the query.

How do we find out all department names that start with M? To do this we use the wildcard '%'. % stands for any character or characters. When you use % you need the SQL word LIKE. To display all departments that start with M we type:

## SELECT \*

**FROM** department

**WHERE** Name LIKE 'M%';



1. Type the SQL query to return the first and last names and department id of all employees who earn less than $55000.

# Multiple Conditions

Sometimes we may need to filter the result set by having more than one condition met. For example, if we wish to list all the Departments that start with M and whose manager ID is 1

## SELECT \*

### **FROM** department **WHERE** Name like 'M%' AND ManagerID = 1;

Both conditions must be true, and in this case only 1 row is returned:

A screenshot of a cell phone  Description automatically generated

However if we change the AND to OR the result set changes. Two rows are returned. When we use an OR condition, only one condition need be true for a row to be returned.

## SELECT \*

**FROM** department **WHERE** Name like 'M%' OR ManagerID = 1;

The query lists all departments starting with M, as well as all departments where the ManagerID is equal to 1.

A screenshot of a cell phone  Description automatically generated

### In MySQL Workbench show the metadata about the Department and Employee tables

DESC Department; DESC Employee;

# Maths Conditions

The table below summarises the conditions and their interpretation in SQL:

|  |  |  |  |
| --- | --- | --- | --- |
| Condition | Meaning | Example | Explanation |
| > | Greater Than | salary > 55000 | value is 55001 or higher |
| < | Less Than | salary < 45000 | value is 44999 or lower |
| = | Equal to | lastname = 'Underwood' | must match exactly |
| >= | Greater than or equal to | salary >= 45000 | value is 45000 or higher |
| <= | Less than or equal to | salary <= 45000 | value is 45000 or lower |
| != <> | Not equal to | name != 'Torch' name <> 'Torch' | can be any value except Torch |

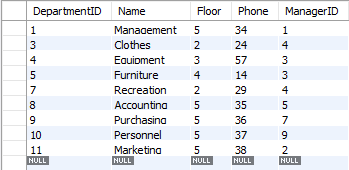
*Table 1: Conditions and their behaviour in SQL*

To select all departments that are above the first floor we would type

## SELECT \*

**FROM** department

**WHERE** Floor > 1;



Or find out which departments are NOT on the fifth floor

**SELECT** Name, Floor **FROM** department **WHERE** Floor != 5;

OR

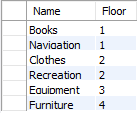
**SELECT** Name, Floor **FROM** department **WHERE** Floor <> 5;

*Note that != and <> both mean 'Not Equal To'*

# ORDER BY

We can order the result set by any column (it does not have to be in the SELECT clause). The ORDER BY forces the result set to be ordered by the values of one or more columns.

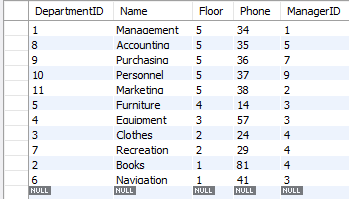
**SELECT** Name, Floor **FROM** department **WHERE** Floor != 5 **ORDER BY** Floor;



The default sort order is from the smallest value to largest (1-10 or A-Z). You can explicitly state this by typing ASC (short for Ascending order). To sort from largest value to smallest you would enter DESC (short for Descending order).

## SELECT \*

**FROM** department

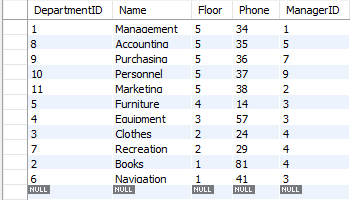
**ORDER BY** Floor DESC;

You can order by more than one column and in different order for each column:

## SELECT \*

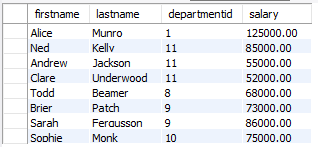
**FROM** department

**ORDER BY** Floor DESC, DepartmentID ASC;



### Type the SQL query that lists the first name, last name, departmentid and salary of all employees who work in DepartmentID 11 AND who earn greater than 55000.

Then change the AND to OR and note the difference in the result set. Notice Clare Underwood's department and salary

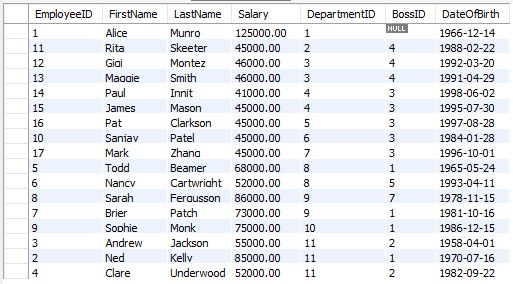


### Type the SQL query that returns the firstname, lastname, salary and departmentid of all employees who earn 45,000 or more. Order the results from highest earner to lowest

Your result set should look like this



### Type the SQL query that returns all rows and columns in the employee table. Order

the result set by departmentid then alphabetically by employee’s lastname.

# LIMIT

We can limit the number of rows in the result set by using the word LIMIT and specifying an integer after the LIMIT word.

A screenshot of a cell phone  Description automatically generated**SELECT** Name **FROM** department **WHERE** Floor = 5

**ORDER BY** Name ASC

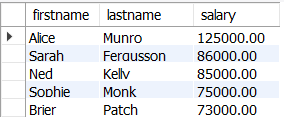
**LIMIT** 2;

### Type the above query and note the two rows returned. Change the ORDER BY from ASC to DESC and rerun the query. Is the result set different? If so, why are they different?

*Hint: to see the difference remove the LIMIT.*

### Type the query that returns the first name, last name and salary of the five highest salary earners across the whole Department store.

Your result set should look like this



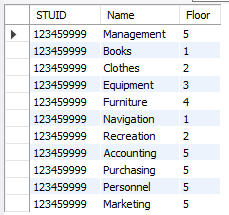
# Adding your Student ID to the result set

### Write the following SQL statement and execute it.

**SELECT '123459999' as STUID,** Name, Floor

**FROM** department;

(Note in the above statement 123459999 should be replaced with your Student ID) Notice how each row contains your student id, department name and floor.



To display all columns:

**SELECT '123459999' as STUID,** department.\*;

**FROM** department;

(Note in the above statement 123459999 should be replaced with your Student ID)

# FUNCTIONS

Functions are mathematical and scientific calculations that are performed automatically by the database engine. There are several function types across all database data types. The most common functions we use are COUNT, MAX, MIN. The full list of Functions you can use in MYSQL are found [here in the MySQL reference](https://dev.mysql.com/doc/refman/5.7/en/func-op-summary-ref.html) [manual](https://dev.mysql.com/doc/refman/5.7/en/func-op-summary-ref.html)

To find out how many departments there are we can use the COUNT() function. Functions must be given something to act on which can be a column, or all columns using the wild card \*

E.g.

**SELECT** COUNT(\*)

**FROM** Department;

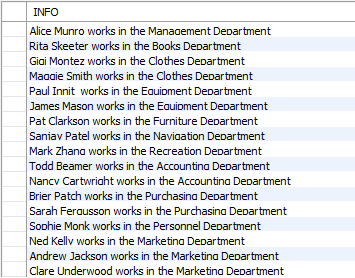
**SELECT** COUNT(Name)

**FROM** Department;

### **SELECT** CONCAT(FirstName,' ',LastName , ' works in the ' , Department.Name, ' Department') AS INFO

**FROM** EMPLOYEE

**NATURAL JOIN** DEPARTMENT;



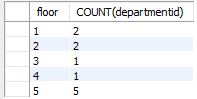
### Type the SQL query to find the total number of employees in the employee table

Your result set should look like this

A screenshot of a cell phone  Description automatically generated

# GROUP BY

Sometimes we want to group the function by a particular attribute. For example to find out the number of each departments on each floor of the department store we would type:

**SELECT** floor, count(departmentid)

**FROM** DEPARTMENT

**GROUP BY** floor;

We use the GROUP BY keyword when aggregate functions are with a column that does not aggregate the rows. We must group by the non aggregated column or columns to ensure the full result set is returned. Thus in the above example we GROUP BY *floor.*

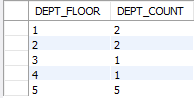
*Try this: Remove the GROUP BY keyword and notice the difference in the query output*

# Alias

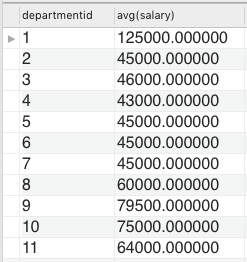
We can also alias the columns to make the output make more sense to the reader. Then use that alias within the query

### **SELECT** floor as DEPT\_FLOOR, COUNT(departmentid) AS DEPT\_COUNT

**FROM** DEPARTMENT **GROUP BY** DEPT\_FLOOR **ORDER BY** DEPT\_FLOOR;



### A screenshot of a cell phone Description automatically generatedType the SQL query to find how mdany employees work in each department

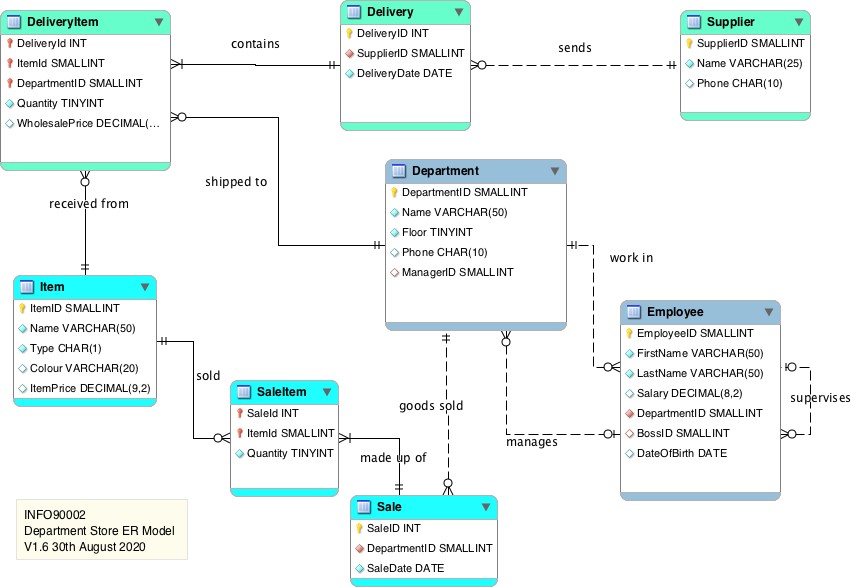
1. Type the SQL query to find each department’s average salary?

### Type the SQL query that finds what department has the highest salary?

1. Type the SQL query that finds the department with the lowest salary?

**END OF WORKSHOP**

Appendix. New department Store Physical ER Model



*Figure A1: The department store schema*