Week 02: SQL

Go to link: http://www.sqlfiddle.com

Input Data

/*	Comment with more than one line
*/

```
CREATE TABLE Employee (
	EMPLOYEE_ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
	FIRST_NAME CHAR(25),
	LAST_NAME CHAR(25),
	SALARY INT(15),
	JOINING_DATE DATETIME,
	DEPARTMENT CHAR(25)
);

INSERT INTO Employee
	(EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY, JOINING_DATE,
DEPARTMENT) VALUES
		(001, 'Neville', 'Longbottom', 100000, '14-02-20 09.00.00', 'HR'),
		(002, 'Ronald', 'Weasley', 80000, '14-06-11 09.00.00', 'Admin'),
		(003, 'Hermoine', 'Granger', 300000, '14-02-20 09.00.00', 'HR'),
		(004, 'Harry', 'Potter', 500000, '14-02-20 09.00.00', 'Admin'),
		(005, 'Severus', 'Snape', 500000, '14-06-11 09.00.00', 'Admin'),
		(006, 'Luna', 'Lovegood', 200000, '14-06-11 09.00.00', 'Account'),
		(007, 'Draco', 'Malfoy', 75000, '14-01-20 09.00.00', 'Account'),
		(008, 'Minerva', 'Mcgonagall', 90000, '14-04-11 09.00.00', 'Admin');

CREATE TABLE Bonus (
	EMPLOYEE_REF_ID INT,
	BONUS_AMOUNT INT(10),
	BONUS_DATE DATETIME,
	FOREIGN KEY (EMPLOYEE_REF_ID)
		REFERENCES Employee(EMPLOYEE_ID)
      ON DELETE CASCADE
);

INSERT INTO Bonus
	(EMPLOYEE_REF_ID, BONUS_AMOUNT, BONUS_DATE) VALUES
```

```
                (001, 5000, '16-02-20'),
                (002, 3000, '16-06-11'),
                (003, 4000, '16-02-20'),
                (001, 4500, '16-02-20'),
                (002, 3500, '16-06-11');
CREATE TABLE Title (
        EMPLOYEE_REF_ID INT,
        EMPLOYEE_TITLE CHAR(25),
        AFFECTED_FROM DATETIME,
        FOREIGN KEY (EMPLOYEE_REF_ID)
                REFERENCES Employee(EMPLOYEE_ID)
      ON DELETE CASCADE
);

INSERT INTO Title
        (EMPLOYEE_REF_ID, EMPLOYEE_TITLE, AFFECTED_FROM) VALUES
 (001, 'Manager', '2016-02-20 00:00:00'),
 (002, 'Executive', '2016-06-11 00:00:00'),
 (008, 'Executive', '2016-06-11 00:00:00'),
 (005, 'Manager', '2016-06-11 00:00:00'),
 (004, 'Assistant Manager', '2016-06-11 00:00:00'),
 (007, 'Executive', '2016-06-11 00:00:00'),
 (006, 'Lead', '2016-06-11 00:00:00'),
 (003, 'Lead', '2016-06-11 00:00:00');
```

Question 1

Select employees from the ‘Employee’ table that belong to the ‘Admin’ department and the result should be displayed in ascending order of salary.

```
SELECT * FROM Employee
WHERE DEPARTMENT = 'Admin'
ORDER BY SALARY ASC;
```

Question 2

Print the complete name and department of the Employees ordered in ascending order by the last name followed by the first name.

Hint: Harry Potter should be printed as Potter Harry

```
SELECT CONCAT(LAST_NAME, ' ', FIRST_NAME) AS Name, DEPARTMENT FROM Employee ORDER BY Name ASC;
```

Question 3

Write an SQL query to print details of the Employees whose FIRST_NAME ends with ‘e’.

```
SELECT * FROM Employee WHERE FIRST_NAME LIKE '%e';
```

Question 4

Fetch the different department names.

```
SELECT DISTINCT DEPARTMENT FROM Employee;
```

Question 5

Find the average salary of employees in each department.

```
SELECT DEPARTMENT, AVG(SALARY) AS AverageSalary FROM Employee GROUP BY DEPARTMENT;
```

Question 6

Select the rows of all the employees who joined in the 6th month of 2014.

```
SELECT * FROM Employee WHERE JOINING_DATE LIKE '2014-06%';
```

Question 7

Extract the amount of money spent by the company in paying salary and bonus.

```
SELECT SUM(SALARY) AS Compensation FROM Employee
UNION
SELECT SUM(BONUS_AMOUNT) FROM Bonus;
```

Question 8

Select employee details who earn learn less than 100000 and have a bonus greater than 3000

```

```

Common Queries and their “variants”

Accessing all values:
SELECT * FROM TableName

Altering a table -
ALTER TABLE TableName ADD column decimal(3,2);
ALTER TABLE Student ADD gpa decimal(3,2);
ALTER TABLE TableName DROP column;

Dropping a table -
DROP TABLE TableName;
DROP TABLE Student;

Insert Data -
INSERT INTO TableName values(d1, d2, d3, …);
INSERT INTO Student values (‘sb1234’, ‘Sirius’, ‘CE’);

INSERT INTO TableName (col1, col2, ...) values (val1, val2, ...);
INSERT INTO Student (student_id, first_name) values (‘msb456’, ‘Slughorn’);

Update Data -
UPDATE TableName
SET col_name = 'value'

Condition based update -
UPDATE TableName
SET col_name = 'value’, col_2 = ‘xyz'
WHERE col_x = 'some_val’ OR col_y = 'something';

UPDATE Student
SET major = 'CS'
WHERE student_id = ‘cc2157';

Delete Row -
DELETE from Student
where student_id = 'sb1234';

Select -
SELECT * FROM TableName;

SELECT col1, col2, …
FROM TableName;

```
SELECT col1,col2,…coln
FROM TableName
ORDER BY col1 DESC, coln ASC;

SELECT col1,col2,…coln
FROM TableName
WHERE col_y = ‘something’
ORDER BY col1 DESC/ASC, coln ASC/DESC
LIMIT n;                    // Prints only the first ‘n’ values

SELECT col1, col2, …
FROM TableName
WHERE col_x IN (‘sb1234’, ‘cc2157’);

SELECT * from TableName
WHERE col_name like ‘_A%’;
--WildCard % ->any number of char, _ -> just one char                    // -- Means comment
//_A% means the first character has to be A and doesn’t matter what is after that
```

Alias -

```
SELECT col_name AS abc from table_name;
```

Select Distinct -

```
SELECT distinct col_name from table_name;
```

Change Text presentation -

```
SELECT upper(col_name) from table_name;
```

Print Selected text from column -

```
SELECT substring(col_name,start,stop) from table_name;
```

Print trimmed text -

```
SELECT TRIM (col_name) from table_name;
```

Print replaced Text -

```
SELECT REPLACE (col_name, 'original', 'new') from table_name;
```

Print combined columns -

```
SELECT CONCAT (col_1, ' ‘, col_2) as final_col from table_name;
```

Union -

```
SELECT first_name as some_name from TableName
UNION
SELECT last_name from TableName;
```

Joins -

```
SELECT a.Col1, a.Col2, b.Col1, ...
FROM TableName1 a
JOIN TableName2 b
ON a.Col_m = b.Col_n;
```

Nested Queries -

```
SELECT a.col1, a.col2, a.col3, b.col1, b.col2
FROM TableName1 a
WHERE a.c_a in      (
                    SELECT a.c1 from TableName2 b where [condition1])
AND [condition2];
```