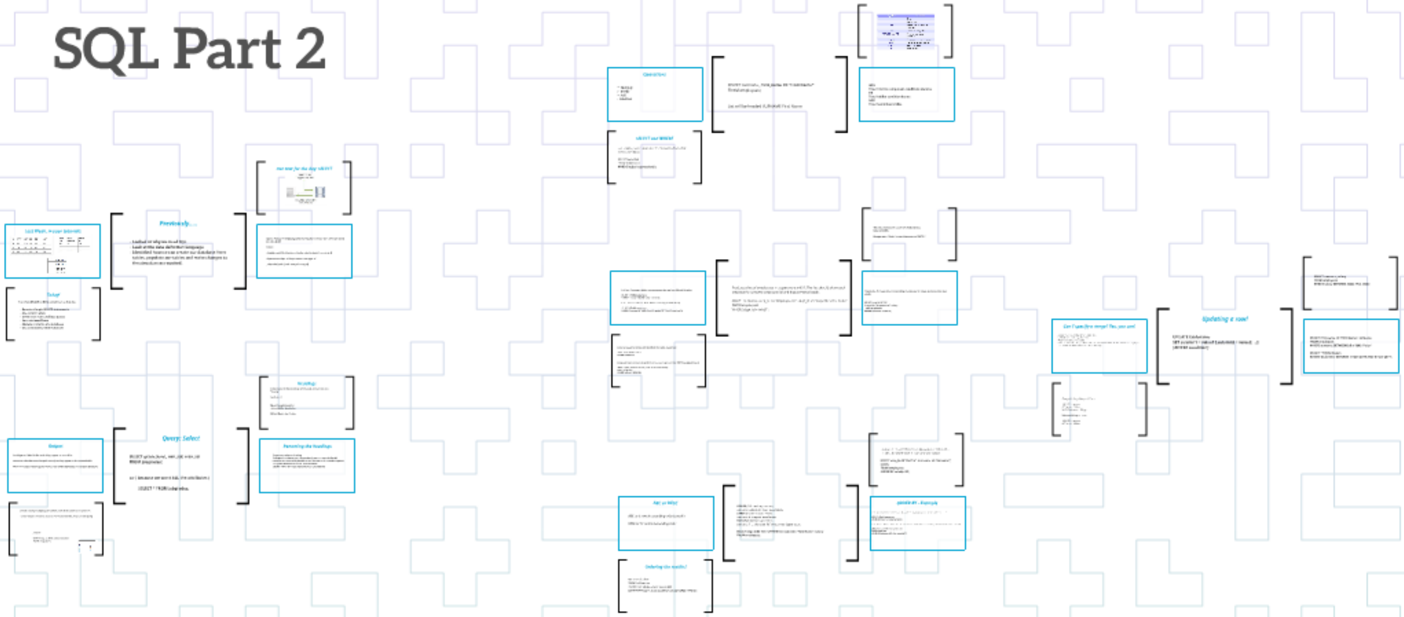
SQL Part 2



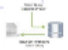
SQL Part 2

Last Week, in your tutorials

Previously.....

- Looked at why we need SQL
- Look at the data definition language
- Identified how we can create our database from tables, populate our tables and make changes to the structures as required.

Our Star for the day: SELECT



[illegible]

Today!

[illegible]

Headings

[illegible]

Query: Select

SELECT grade,level, min_sal, max_sal
FROM jobgrades;

Or (because we want ALL the attributes)

SELECT * FROM jobgrades;

Output

[illegible]

Renaming the headings

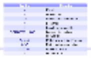
[illegible]

Operations

- Multiply
- Divide
- Add
- Subtract

SELECT surname , first_name AS "First Name"
FROM employees;

List will be headed SURNAME First Name



AND
[illegible]
OR
[illegible]
NOT
[illegible]

SELECT and WHERE

[illegible]

Produce a list of employees in department [illegible]

SELECT surname, emp_id AS "Employee ID", dept_id AS "Department Code"
FROM employees
WHERE dept_id = [illegible]

ASC or DESC

ASC [illegible]

DESC [illegible]

Ordering the results!

[illegible]

ORDER BY - Example

[illegible]

Can I specify a range? Yes, you can!

[illegible]

Updating a row!

UPDATE tablename
SET column1 = value1 [,column2 = value2, ...]
[WHERE condition];

Previously.....

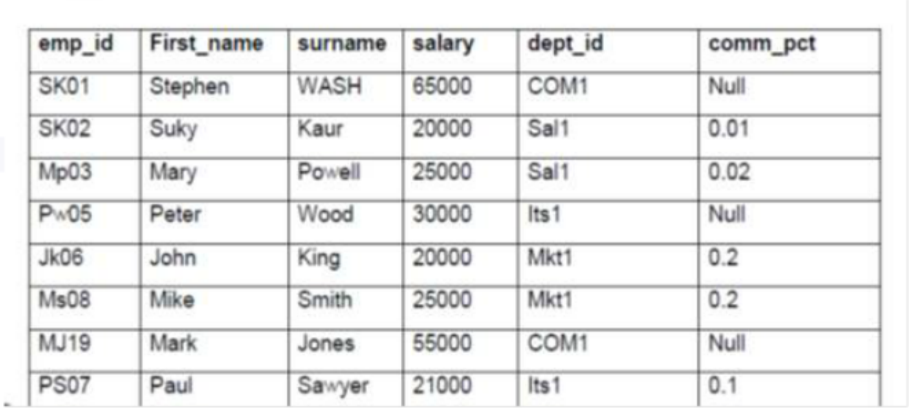
- **Looked at why we need SQL**
- **Look at the data definition language**
- **Identified how we can create our database from tables, populate our tables and make changes to the structure as required.**

Today!

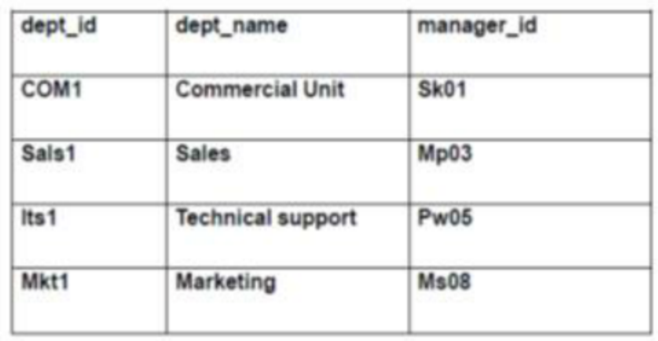
You should with a little practice be able to:

- **Execute Simple SELECT statements**
- **Use column alias**
- **Limit rows retrieved by a query**
- **Sort retrieved Rows**
- **Update contents of a database**
- **Use Case-conversion Functions**

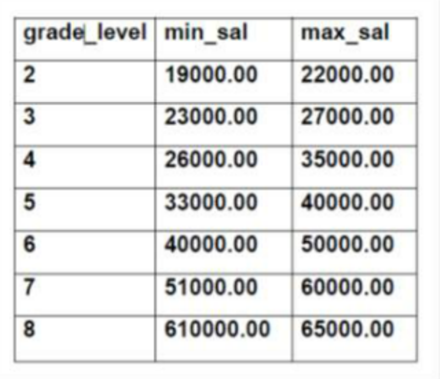
Last Week, in your tutorials



emp_id	First_name	surname	salary	dept_id	comm_pct
SK01	Stephen	WASH	65000	COM1	Null
SK02	Suky	Kaur	20000	Sal1	0.01
Mp03	Mary	Powell	25000	Sal1	0.02
Pw05	Peter	Wood	30000	Its1	Null
Jk06	John	King	20000	Mkt1	0.2
Ms08	Mike	Smith	25000	Mkt1	0.2
MJ19	Mark	Jones	55000	COM1	Null
PS07	Paul	Sawyer	21000	Its1	0.1

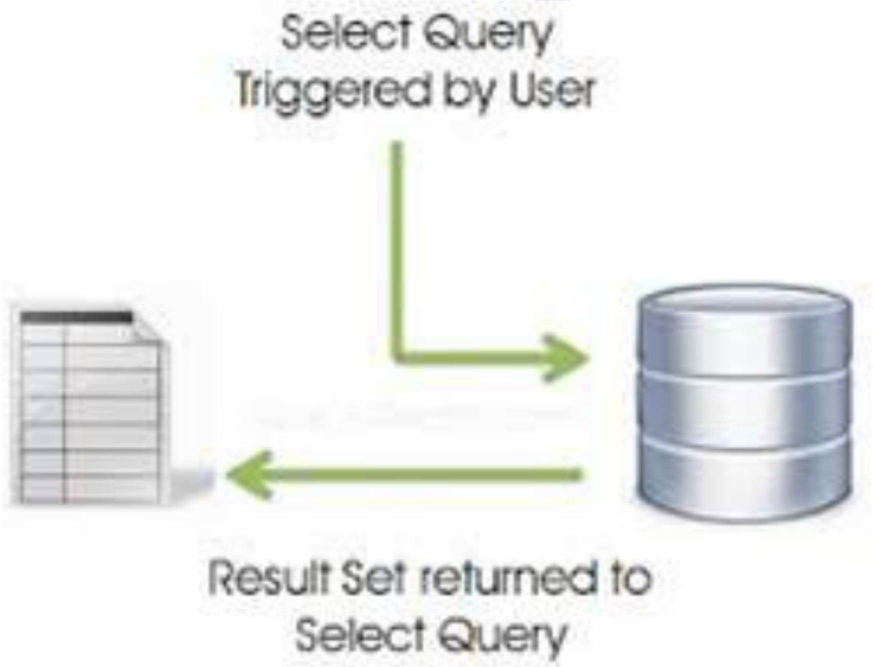


dept_id	dept_name	manager_id
COM1	Commercial Unit	Sk01
Sals1	Sales	Mp03
Its1	Technical support	Pw05
Mkt1	Marketing	Ms08



grade_level	min_sal	max_sal
2	19000.00	22000.00
3	23000.00	27000.00
4	26000.00	35000.00
5	33000.00	40000.00
6	40000.00	50000.00
7	51000.00	60000.00
8	610000.00	65000.00

Our Star for the day: SELECT



Query: Produce a list of job grades showing the minimum and maximum salary for each grade

Tables

employees(id, First_name, surname, salary, dept_id, comm_pct)

departments(dept_id, depat_name, manager_id)

Jobgrades(grade_level, min_sal, max_sal)

Query: Select

```
SELECT grade_level, min_sal, max_sal
FROM jobgrades;
```

Or (because we want ALL the attributes)

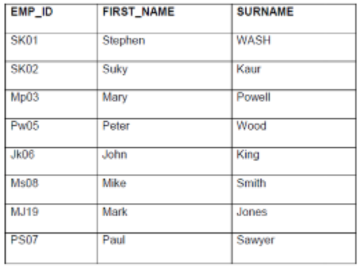
```
SELECT * FROM jobgrades;
```

Create a query to display all staff ids, their first names and surnames.

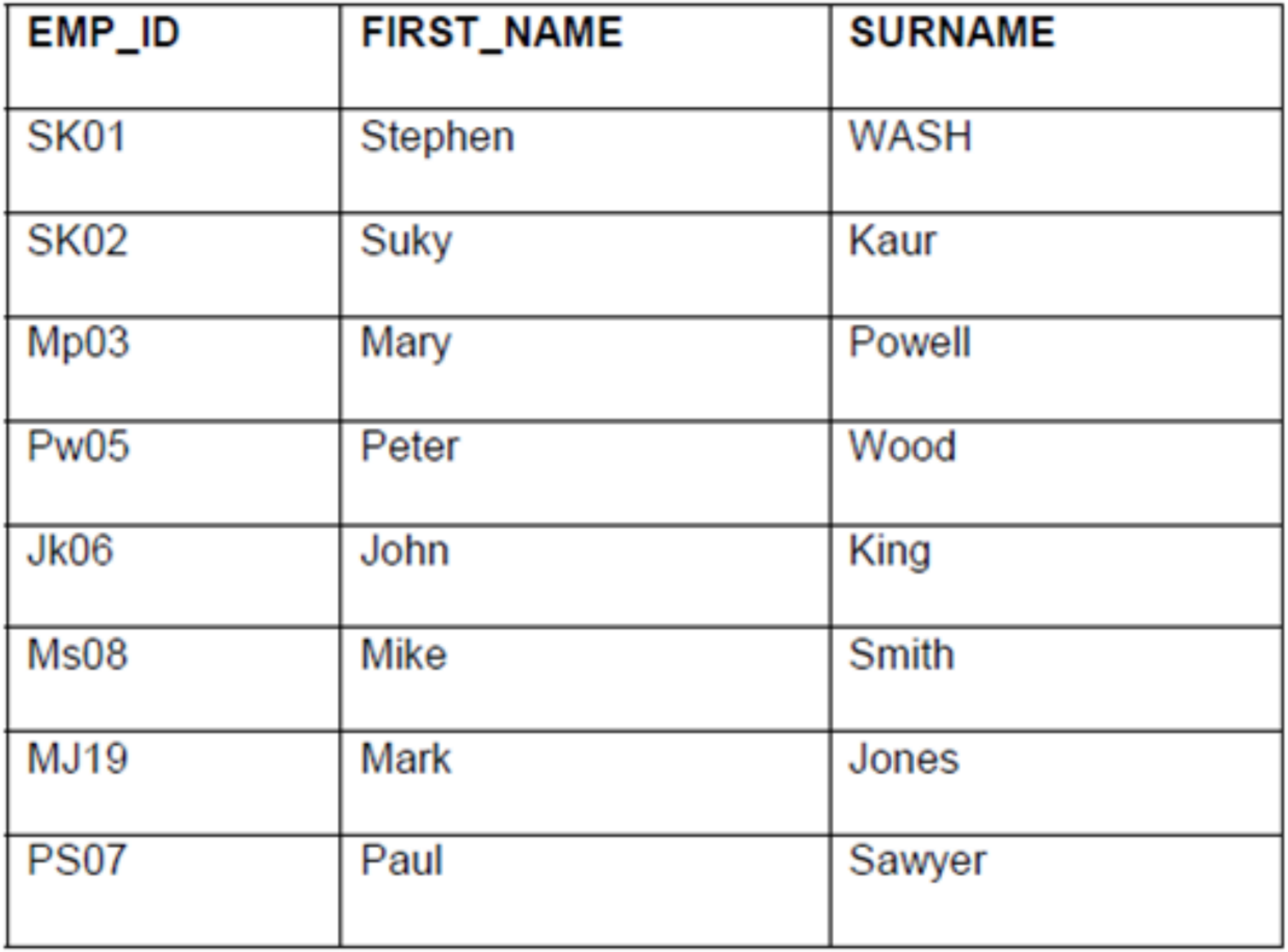
Decide table(s) required and columns from that table(s) to create query.

Answer:

```
SELECT emp_id,first_name,surname
FROM employees;
```



EMP_ID	FIRST_NAME	SURNAME
SK01	Stephen	WASH
SK02	Suky	Kaur
Mp03	Mary	Powell
Pw05	Peter	Wood
Jk06	John	King
Ms08	Mike	Smith
MJ19	Mark	Jones
PS07	Paul	Sawyer



EMP_ID	FIRST_NAME	SURNAME
SK01	Stephen	WASH
SK02	Suky	Kaur
Mp03	Mary	Powell
Pw05	Peter	Wood
Jk06	John	King
Ms08	Mike	Smith
MJ19	Mark	Jones
PS07	Paul	Sawyer

Output

Headings are listed in the order they appear on selectlist.

Names on selectlist must be spelt exactly as they appear in the named table.

When “*” is used, headings are in the order of the attributes in the table structure

Headings

Some names in the headings of list produced are not user friendly:

Eg. Emp_id

What if requirement is?
List should be headed as:

ID First Name Last Name

Renaming the headings

- Renames a column heading
- Follows the column name immediately can be preceded by AS
- Must be surrounded by double quotation marks if it contains spaces or special characters or it is case-sensitive
- Useful with arithmetic expressions or calculations

```
SELECT surname , first_name AS “First Name”
FROM employees;
```

List will be headed SURNAME First Name

SELECT and WHERE

For limiting rows returned restricts rows to those that meet a condition

```
SELECT selectlist
FROM tablename
WHERE logical expression(s);
```

Operations

*	Multiply
/	Divide
+	Add
-	Subtract

Symbol	Operator
=	Equal
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
BETWEEN … AND …	Between two values (inclusive)
IN (set)	Match any of a list of values
LIKE	Match a character pattern
IS NULL	Is a NULL value
!=	Not equal to

AND

True if its two component conditions are true

OR

True if either condition is true

NOT

True if condition is false

Produce a list of employees in department mk01. The list should show each employee’s surname employee id and departmental code.

```
SELECT surname, emp_id AS “Employee ID”, dept_id AS “Departmental Code”
FROM employees
WHERE dept_id = ‘mk01’;
```

List of all customer details with CustID of 10 in table Customers:

```
SELECT * FROM Customers
WHERE CustID=10;
```

List grade level and max salary where min salary is greater than 35000 in jobgrades table

```
SELECT grade_level AS “Grade”, max_sal AS “Max Salary”
FROM jobgrades
WHERE min_sal> 35000.00;
```

List from Customers table those customers who are from UK and Coventry

```
SELECT * FROM Customers
WHERE Country=‘UK'AND City=‘Coventry';
```

List customers who are from UK and (Coventry or Sunderland)

```
SELECT * FROM Customers
WHERE Country=‘UK' AND (City=‘Coventry' OR City=‘Sunderland');
```

Must be enclosed in a pair of single quotes
Case sensitive

Comparison: 'Smith' is not the same as 'SMITH'

Display the ID, Department and salary of employer(s) whose surname(s) is (are) Smith:

```
SELECT emp_id AS “ID”,
dept_id AS “Department”, salary
FROM employees
WHERE surname = ‘Smith’;
```

LOWER (Column/expression)
Converts value into lower case letters
UPPER (Column/expression)
Converts into upper case letters
INITAP (Column/expression)
Converts first character (left most) into Upper Case:

```
SELECT emp_id AS “ID”, UPPER(first_name) AS "First Name", salary
FROM employees;
```

Ordering the results!

```
SELECT selectlist
FROM tablename
[WHERE conditional expression(s)]
[ORDER BY {col1, expr, position_of_col} [ASC|DESC]];
```

ASC or DESC

ASC sorts rows in ascending order (default)

DESC sorts rows in descending order

Produce a list of staff in ascending order of salary; the list should include staff id, surname and salary:

```
SELECT emp_id AS “staff id”, surname  AS “Surname”,
salary
FROM employees
ORDER BY salary ASC;
```

ORDER BY - Example

Select all records from a Customer table and order by Country and CustomerName attributes:

```
SELECT * FROM Customers
ORDER BY Country, CustomerName;
```

Select all emp_id, first_name and surname from a Customer table and order by Surname (A-Z), first_name (Z-A)

```
SELECT emp_id, first_name, surname
FROM employees
ORDER BY surname ASC, first_name DESC;
```

Updating a row!

```
UPDATE tablename
SET column1 = value1 [,column2 = value2, ...]
[WHERE condition];
```

Change Mr Kings Salary to £25000

```
UPDATE employees
SET salary = 25000
WHERE surname = “King”;
```

What would happen here?

```
UPDATE employees
SET salary = 25000;
```

Can I specify a range? Yes, you can!

- Range has a lower limit and an upper limit
- Lower limit must be specified first
- Values specified are inclusive
- Produce a list of employees that earn salaries between 3000 and 45000. For each employee the list should show his/her surname and salary.

```sql
SELECT surname, salary
FROM employees
WHERE salary BETWEEN 30000 AND 45000;
```

```
SELECT first_name AS “First Name”, surname
FROM employees
WHERE surname BETWEEN ‘John’ AND ‘Peter’

SELECT * FROM Orders
WHERE OrderDate BETWEEN ’01-jan-2014’ AND ’07-jan-2014’;
```

SQL Part 2

Last Week, in your tutorials

Today!

[illegible]

Previously.....

- Looked at why we need SQL
- Look at the data definition language
- Identified how we can create our database from tables, populate our tables and make changes to the structure as required.

Our Star for the day: SELECT

Headings

Query: Select

```
SELECT grade_level, min_sal, max_sal
FROM jobgrades;
```

Or (because we want ALL the attributes)

```
SELECT * FROM jobgrades;
```

Output

Renaming the headings

Operations

SELECT and WHERE

```
SELECT surname , first_name AS "First Name"
FROM employees;
```

List will be headed SURNAME First Name

AND

OR

NOT

```
SELECT surname, emp_id AS "Employee ID", dept_id AS "Department Code"
FROM employees
WHERE dept_id = [illegible]
```

Can I specify a range? Yes, you can!

Updating a row!

```
UPDATE tablename
SET column1 = value1 [,column2 = value2, ...]
[WHERE condition];
```

ASC or DESC

ASC sorts rows in ascending sequence (default)

DESC sorts rows in descending order

Ordering the results!

ORDER BY - Example