Selecting Data

Name:	Class:	Date:

When you want to obtain data from a database you do so by constructing queries.

These queries use SELECT to pull data that matches certain conditions.

A SELECT query can be structured different way:

-- Simplest query that returns all records.

SELECT *fields* FROM *table*;

-- A single condition takes records that only meet the condition (*field* ***operator*** *value)*

SELECT *fields* FROM *table* WHERE *field* ***operator*** *value*

-- A multi condition that uses Boolean to link conditions

SELECT *fields* FROM *table* WHERE *field* ***operator*** *value* ***Boolean*** *field* ***operator*** *value ...*

We can get every field by using *.

SELECT * FROM FROM teachers;

We can state specific fields in the SELECT by naming them:

SELECT first_name, last_name FROM students;

Or get only specific records:

SELECT * FROM students WHERE last_name = “Hull”;

Selecting Data

Task 1

Task 1.1

First we will do a simple query to get all records from the students table.

SELECT * FROM students;
The * means all fields.

Task 1.2

Now we want to reduce the number of fields returned.

Create a new query that only selects the student_number and last_name.

Task 1.3

Next we want to query teachers and only return WHERE the name is Richard.

It is possible to query using multiple conditions by including Boolean (AND, OR).

Before we do this let us make sure that there are some similar records in teachers.

Task 1.4

Add the following records into teachers.

7, 'Richard', '2000-06-01', 90000

8, ‘Zak’, ‘2001-05-01’, 89000

Task 1.5

Write a query that will only bring up the latest teachers names Zak and Richard.

Selecting Data

Below are a set of built-in SQL functions.

They can be used in different parts of a statement to achieve different results.

```
-- Date Functions
SELECT CURRENT_DATE() AS 'Current Date';
SELECT CURRENT_TIME();
SELECT CURRENT_TIMESTAMP();
SELECT NOW();
SELECT MONTHNAME(CURDATE());
SELECT YEAR(CURDATE());
SELECT DAYNAME(CURRENT_DATE());
SELECT CURDATE();
SELECT DAY(CURDATE());


-- Math functions
COUNT()
SUM()
PI()
RAND()

-- String Functions
SELECT TRIM('                Richard                ') AS Name;
SELECT LTRIM('                 Richard                 ') AS Name;
SELECT RTRIM('                 Richard                 ') AS Name;
SELECT UPPER('zak');
SELECT UCASE('zak pArdis');
SELECT LOWER('ZAK');
SELECT LCASE('Richard hUNt');
SELECT REVERSE('Google');
SELECT SUBSTRING('Zak Pardis',3);
SELECT SUBSTRING('Zak Pardis',3,4);
```

Selecting Data

```
-- Administration Functions
SELECT CURRENT_USER();
SELECT VERSION();
SELECT SESSION_USER();
SELECT CONVERT(10, CHAR);
```

You can also combine the functions to create more complex expressions.

```
SELECT CONCAT(DAYNAME(CURDATE()),", ",DAY(CURDATE()),' ', MONTHNAME
(CURDATE()), ' ',YEAR(CURDATE())) AS Date;
```

We can also use them while selecting data.

```
-- Identifies duplicates
-- This statement has a COUNT in both SELECT to create a field (the alias
city_name), and HAVING to count duplicate entries.
SELECT city_name, COUNT(city_name) AS duplicate_count
FROM cities
GROUP BY city_name
HAVING COUNT(*) >1;
```

Task 2

Task 2.1

Use a source of data such as Mockaroo.com to generate at least 100 piece of student data.

Now you will COUNT() how many students share a first name, ensure you add an alias field to show the count of names.

Selecting Data

The HAVING part of the statement is where we can do mathematics such as COUNT, SUM, RAND().

This cannot be done in the WHERE section of a statement as WHERE only compares values.

Fore example we can do this:

```
SELECT class_start , class_id FROM classes WHERE class_start > NOW();
```

We can also use a function to temporarily transform data from a field.

In this example we take only the YEAR() from start date and return those dates greater than 2000.

```
SELECT * FROM subjects WHERE YEAR(start_date) > '2000';
```

Task 2.2

Create a query that returns the trainer_name and trainer_dob for teachers who were born after the year 2000.

Task 2.3

Create a query that sums all the wages of all teachers.

- Hint: This will end up with a single field and row with a combined wage.

Selecting Data

Challenges

Challenge 1

ORDER BY can be used to create alphabetical or numerical order.

SELECT *fields* FROM *table* ORDER BY *field*

Create a query that shows each teacher name in capital letters, their salary and use the ORDER BY to show them in salary order.

Challenge 2

ORDER BY can use ASC and DESC of which ASC (accend is the default).

We can also use both GROUP BY and ORDER BY in the same query.

Create a query that:

Selects the subject_name and counts subject_name from subjects.

Group and order by the subject_name.