Brightlearn Tutorials

Exercise 1: SQL Fundamentals : Lubabalo Sitoma

Questions:

Q1. Select *
FROM Employees;

Q2. Select Distinct (Department)
FROM Employees;

Q3. Select First_name,
last_name,
Salary
FROM Employees
Order By Salary DESC;

Q4. Select First_name,
last_name,
Salary,
FROM Employees
Order By Salary DESC
limit 5;

Q5. Select First_name,
last_name,
Department
FROM Employees
Where Department = 'IT';

Q6. Select First_name,
last_name,
Department,
Salary
FROM Employees
Where Department = 'Finance' AND Salary > 58 000

Q7. Select First_name,
last_name,
Department
FROM Employees
Where Department = 'HR' OR Department = 'Marketing';

Q8. Select First_name,
last_name,
Department
FROM Employees
Where department NOT IN ('IT');

Q9. Select First_name,
last_name,
Department
FROM Employees
Where Department IN ('HR', 'IT', 'FINANCE');

Q10. Select First-name,
last-name,
Department,
Salary,
City
FROM Employees
Where Department = 'IT' AND Salary > 50000 AND
City = 'New York';

Q11. Select First-name;
last-name,
Department,
Salary
FROM Employees
Where department = 'Finance OR Department = 'marketing' AND
Salary > 52 000
ORDER By Salary DESC;

Q12. Select Distinct (City);
Department
FROM Employees
Where Department NOT IN ('IT', 'HR');

Q13. Select First_name;
last_name,
Salary,
Hire_Date.
FROM Employees
Where Department NOT IN ('Finance') AND salary > 50000
ORDER By Hire_Date ASC;

Q14. Select First_name;
last_name;
Department;
City.
FROM Employees
Where City = ('Chicago', 'Los Angles') AND Department =
IT OR Department = Marketing.
limit 3;

Exercise 2: SQL AGGREGATE FUNCTIONS & SQL OPERATORS : Lubabalo Sitama

Questions:

G1.

```
Select Distinct ('Department')
FROM Students;
```

G2.

```
Select Department,
       AVG ( AGE) AS AVG_AGE
FROM Student
GROUP By Department;
```

G3.

```
Select Department;
       COUNT (*) AS Student_Count
FROM Students
GROUP By Department
Having Count (*) > 1;
```

G4.

```
Select Student_ID,
       name,
       Age,
       Department,
FROM Students
Where age between 21 AND 23;
```

Q5. Select Student_ID;
name,
age,
Department
FROM Students
Where Department = ('HR') OR Department =('HR') AND
Age > 21;

Q6. Select Department;
Sum (Credits) AS total_Credits
FROM courses
GROUP By Department
Having sum (Credits) > 8;

Q7. Select course_ID,
course_name,
Department,
Credits
FROM courses
Where Credits < 4

Q8. Select course_ID,
course_name,
Credits.
FROM courses
Order By Credits DESC
limit 3;

Q9. Select MAX (grade) AS max_grade,
MIN (grade) AS min_grade,
AVG (grade) AS avg_grade.
FROM Enrollments;

Q10. Select course_ID,
Count (Enrollment) AS enrollment_count,
FROM Enrollment
GROUP By course_ID;

Q11. Select Department
SUM (Salary) AS total_Salary,
SUM (Bonus) AS total_Bonus
FROM Salaries
GROUP By Department

Q12. Select Department,
AVG (Salary) AS AVG_Salary
FROM Salaries
GROUP By Department
Having Salary > 50000;

Q13. Select Employee_ID,
name,
Salary,
bonus,
SUM (Salary AND Bonus) AS total_compensation
FROM Salaries

Q.14. Select Department,
Sum (Budget) AS total - budget,
AVG (Budget) AS AVG - budget
FROM Prgects
where avg - budeget > 70000;

OR

Q.14 Select Department;
sum (Budget) AS total - budget,
AVG (Budget) AS AVG - budget
FROM ~~Prgects~~
GROUP By department
Having avg - budget > 70000;

Q.15. Select Project - ID,
project - name,
Department,
Budget
FROM Prgects
Where Budget between 50000 AND 120000 AND
department != Marketing;

Exercise 3: SQL CASE Statements

Lubabalo Sitama

Questions:

Q1
```
Select Product_name,
       Price
Case   When Price > 1000 then 'Exepensive',
       When Price between 100 AND 1000 Then 'Mid-range',
       When Price < 100 then 'budget'
END as price_category
FROM products;
```

Q2
```
Select customer_name,
       amount
Case   When Amount ≥ 1000 Then 'High value';
       When Amount between 300 AND 999,99 Then
       'Medium value',
       When Amount < 300 Then 'low value'
END as order_value_category
From orders;
```

Q3
```
Select emp_name,
       department,
       salary
CASE   When department = 'IT' AND Salary > 80000 Then
       'Senior IT',
       When department = 'HR' AND salary > 55000 THEN
       'Experienced HR',
```

```
ELSE 'Staff'
END AS position_level
FROM Employees
```

Q4.

Note: Please replace the grades crossed out with 'score'

```
Select student_name,
       score
CASE   When ~~grade~~ score ≥ 90 then 'A',
       When ~~grade~~ score between 80 AND 89 then 'B',
       When ~~grade~~ score between 70 AND 79 then 'C',
       When ~~grade~~ score between 60 AND 69 then 'D',
       When ~~grade~~ score < 60 then 'F'
END AS grade
FROM Students;
```

Q5.

```
Select delivery_ID,
       delivery_time_mintes
CASE   When delivery_time_mintes ≤ 30 mins Then 'fast',
       When delivery_time_minutes between 31 AND 60 then
       'on-time',
       When delivery_time_minutes > 60 Then 'late'
END AS performance
FROM deliveries;
```

Q6.

```
Select Issue_type,
       priority
Case   When priority = 3 then 'High',
       When priority = 2 then 'medium',
       When priority = 1 then 'low'
END AS priority_label
```

```
FROM    tickets;
```

Q7. ~~Select Student_ID;~~

```
Select (*)
        (days_present)/total_days) * 100
END AS attendance_percentage
FROM Attendance
-- This code will give you the percentages for all
 rows
-- This will then be followed by the code to solve
 the question

Select Studen_ID;
        attendance_percentage
CASE    when atterdance_percentage ≥ 90 % then 'Excellents';
        When attendance_percentage between 75 AND 89 then
        'Good';
        When attendance_percentage ≤ 75 then 'Needs
        Improvement'
END     AS  attendance_Status
FROM    Attendance;
```

Q8.
```
Select product_ID;
        stock_qty
CASE    when stock_qty = 0 then 'out of stock';
        when stock_qty between 1 AND 5 then 'Low Stock';
        when stock_qty = 5 then 'In Stock'
END AS  stock_status
```

```
FROM products_inventory;
```

Q9.
```
Select subject,
       enrolled_students
Case when enrolled_students >= 25 then 'large',
     when enrolled_students between 10 AND 24 then 'medium';
     When enrolled_students < 10 then 'small'
END AS class_size_category
FROM Classes;
```

Q10.
```
Select payment_ID,
       payment_method,
       amount
CASE when payment_method = 'cash' AND amount >= 200 Then
     'Eligible for Discount',
ELSE 'Not Eligible'
END AS discount_eligibility
FROM payments;
```

Exercise 4: SQL JOINS Exercise: Lubabalo Sitoma

Questions:

Q1. Select Student_ID,
Student_name,
Grade
FROM Students AS A
INNER JOIN Grades AS B
On. A. Student_ID = B. Student_ID;

Q2. Select emp_ID,
Emp_name,
dept_name
From employees AS A
Left JOIN departments AS B
On. A. emp_ID = B. emp_ID;

Q3. Select product_ID,
product_name,
quantity
FROM products AS A
FULL OUTER JOIN sales AS B
On. A. product_ID = B. product_ID;

Advanced Join Practice : Exercise 4

Q4. Select order_ID,
customer_ID,
amount,
customer_name
Case When customer_ID IS NULL Then 'New'
When customer_ID IS NOT NULL Then 'Returning'
END AS customer_type
FROM orders AS A
Left JOIN customers AS B
On A.customer_ID = B.customer_ID;

Q5. Select region_ID,
region_name,
SUM (Amount) AS total_sales
GROUP BY region_ID
FROM Sales AS A
Left JOIN regions AS B
On A.region_ID = B.region_ID;

Q6. Select student_ID,
name,
days-present
CASE When days-present ≥ 18 then 'Excellent',
When days-present between 5 AND 17 then 'Needs Improvement',
When days-present < 5 then 'Poor Attendance',
Else 'No Record'
END AS attendance_status
FROM students AS A

Left JOIN attendance AS B
On. A. Student_ID = B.Student_ID;

Q7. Select project_ID,
name
COUNT(*) AS task_count
Group By name
FROM projects AS A
INNER JOIN tasks AS B
On. A. project_ID = B. project_ID;

Q8 ~~Select cust_ID,~~
~~Order_total~~
~~Case When~~
Select cust_ID,
order_total,
return_total,
CASE When return_total IS NOT Null then 'returned',
ELSE 'Not returned' Note: Forgot to add END AS return_Status
FROM orders AS A
Full outer JOIN returns AS B
On. A. cust_ID = B. cust_ID
Where order_total > 100;

Q9.
```
Select user_ID,
       name,
       COUNT(login_date) AS login_count
FROM users AS A
Left JOIN logins AS B
On A.user_ID = B.user_ID
ORDER by login_count DESC
```

note: forget to add Group by user_ID

Q10.
```
Select teacher_ID,
       teacher_name,
CASE when
Select teacher_ID,
       teacher_name,
       subject_name
CASE when subject_name IS NULL Then 'NO subject Assigned'
END AS subject_name
FROM techers AS A
Left JOIN subjects AS B
On A.teacher_ID = B.teacher_ID
ORDER By teacher_name ASC;
```

Research Assignment 1: Lubabalo Sitoma

Section A:

1. ▸ Rational Database
   ▸ Non-retional Database

2. It is the system which you use to gain access to the database in order to manipulate/manage the data that is saved on the database.

3. A primary key is a unique identifier for each row within a dataset which ensures that each record within a dataset is unique and can be uniquely identified and a foreign key is a column within a table which is a primary key in the main table or another table but allows connectivity or a link between the two table thraigh being a primary key in the main table and a foreign key in the other.

4. Database normalization is the process of organising data and cleaning it, this is important and done in order to make data logical/sensible and free from errors, repetitions and many more. It involves the process of also analysing the dat and breaking it down to smaller, understandable fragments.

5. A schema plays the role of a folder, it defines the way in which rational data is organized.

6. A structed data is data that is consistent, clean and that is stored within tables & rows, it can be manipulated thraigh structed query language, semi-structured data is data that brigdes the gap between structed data & unstructed data because it is consistent to a certain extent, it is moe flexible than structed data, unstructured data on the other hand is data that is not consistent and is not agonized, it is the most flexible type of data and structured query language is the lowest on this type of data.

7. A fact table is mostly used for analysis, it is used to store measurable, quantitative data about business processes, it is a numbers table, but a dimension table contains descriptive attributes/ information, it describes the fact table and mostly has primary keys.

8. A data model is the one that is used to create databases, populate data warehouses, manage data for analytical processing & the implementation of applications that allow users to access information in a meaningful way. They are important because they allow the process of information to be accessable & understandable to business users, so they allow for the simplification of the information.

9. A database is where data is stored, a datawarehouse is a central place where multiple Databases are stored, then a data lake is an example of a relational database it is designed to any type of data whether structured, semi-structured or unstructed.

10. A data mart forms part of a data warehouse but is a focused data warehouse, it stores Databases for a specific line or department so it is very narrow and is only focused on storing information for any one business line but a data warehouse holds data across the entire organization. So a data warehouse is basically broken down into multiple data marts

Section B: SQL & data Processing

11. A query language is a language that is used to create communication with databases for the purpose of data retrieval, managing & manipulation. SQL is most commonly used because of it's user friendliness, it is easy to learn & understand because it is pure english based, secondly it is mainly designed for the manipulation of data saved or stored in rows, columns & tables, thirdly it intergrates well with other coding languages.

12. Indexes are data structures that assist to improve data retrieval. During data querying indexes assist to allow the database to quickly locate what is needed within that query without having to scan through the entire table

13. The database transactions is a series of transactions within a database management system. These transactions are set up in a way that if a single step within the transaction fails the entire transaction fails. So this set-up or put in place in order to protect the data with in the database & ensure that it always remains reliable. ACID properties (Atomicity – Consistency – Isolation – Durability) are 4 principles that gearntee the reliability & validity of database transactions.

14. This is the core property of the Data Management System that handles all the functions that performed on data eg: data storage, data retrieval & data manipulation. The engine is the one the executes the queries so the are components within it that assist to speed up the operations.

15. views: It is a shortcut to use when seeking to look at table from a table, so it essentially works like a quick view or overview.

Store procedures: Is a fundamental where queries are stored in order to be used again at a later stage

Triggers: Triggers act as automatic actions in the database such as when data is being inserted or deleted, triggers can make something else also happen automatically & simaulaously

16. The ETL (Extract, Transform, Load) is the process whereby data is extracted, transformed as in being cleaned, the load on the database but an ELT process loads before transforming the data

17. Batch processing is the process of collecting data & processing it onto the database all at once at a specified time during the day usually during non-operational hours. but Stream processing is creating a seamless streamline to process the data onto the database continaisly as it comes.

18. A JOin combines two table with a common column between the two tables, so it basically uses the primary & foreign key to combine the tables

- INNER JOIN
- Left JOIN
- Right JOIN
- Full OUTER JOIN

INNER JOIN : Retrieve data from the tables where a common match between the tables

```
Select column (S).
From table 1 AS A
INNER JOIN table 2 AS B
On. A common_column = B.common-column;
```

Left JOIN : Retrieve all the data from the base table and match it to the right table.

```
Select column (S)
From table 1 AS A
LeFT JOIN table 2 AS B
On. A. common- column = B. common- column;
```

Right JOIN : Retrieve all the data from the base table and match it to the left table.

FULL OUTER JOIN : Retrieve all the data from both tables

19. It makes sure relationships between tables stay consistent. It is important because it prevents data from getting corrupt

20. Data redundancy is basically having duplicates of datasets within your data, this affects performance in the sense that when performing

queries having alot of rows to go through slows down performance

Section C: Data management AND Analytics Concepts

21. Cloud databases are managed online and the transactions happen online & managed by someone else but on-premise databases are manage and hosted on your own servers

22. Are the set of rules on how data should be handle, basically the control accesibility & establish ways of how the data can be safely kept. It is important in the protection of the data, it controls usage of the data and keeps the data accurate.

23. Data intergrity means that the data is accurate and has been tested, queried & it is reliable and it is at a level where it may also be made accesible & understood by business users. Process are put into place to maintain data integrity, processes such as audits, error checks, validation procedures, encryption, access control & backups.

24. Data quality means that the data is of a high standard, it is reliable & may be easily understood by business users. Data quality is critical because if it is not achieved it may lead to bad decision

& may have cost implications.

25. Data Analyst query data, they clean it to make reliable & easily understandable, they study the data & present common findings & trends based on it so that informed decisions may be made.

26. DBA is manage database performance, security, backups, & ensure data availability. They also handle troubleshooting & optimization.

27. Designing a data pline inwlves defining sources, setting up extraction (ETL/ELT), tranforming data, loading into a target system, & monitoring

28. ▸ Maitaining databases
▸ Ensuring security
▸ Managing backups.

29. ▸ My SQL -
▸ Snowflake -
▸ Microsoft SQL Server -

30. ▸ CSV -
▸ JSON -
▸ Apache Parquet -
▸ XML -