Detailed Solutions to SQL and Database Questions (100 Marks)

Question: 1a. Mention the differences between UNION and UNION ALL.

Solution: UNION combines the results of two queries and removes any duplicate records. UNION ALL also combines results but keeps all duplicate records. Example:

SELECT name FROM TableA UNION SELECT name FROM TableB;

vs.

SELECT name FROM TableA UNION ALL SELECT name FROM TableB;

Question: 1b. Can we use aggregate functions in WHERE clause?

Solution: No, aggregate functions (like SUM, COUNT) cannot be used in the WHERE clause. The WHERE clause filters rows before aggregation, so aggregates must be used in the HAVING clause. Example: SELECT AVG(Salary) FROM Employees WHERE Age > 30 (incorrect).

Question: 1c. Mention the difference between TRUNCATE and ROUND functions.

Solution: TRUNCATE cuts off the decimal without rounding, while ROUND adjusts to the nearest value based on decimal position. Example: TRUNCATE(5.678, 2) results in 5.67, ROUND(5.678, 2) results in 5.68.

Question: 1d. Mention the difference between ISNULL() and IFNULL().

Solution: ISNULL() is a boolean function that checks for NULL values, whereas IFNULL() returns a substitute if the original value is NULL. Example: ISNULL(value), IFNULL(value, 0) where NULL is replaced by 0.

Question: 1e. What is referential integrity?

Solution: Referential integrity ensures that a foreign key correctly references a primary key in another table. This prevents orphaned records, ensuring consistency between tables.

Question: 2a. Explain about EXISTS operator.

Solution: EXISTS is used in subqueries to check if any rows are returned. It's often used to validate data presence, e.g., SELECT * FROM Customers WHERE EXISTS (SELECT * FROM Orders WHERE Customers.id = Orders.CustomerID);

Question: 2b. Difference between WHERE and HAVING.

Solution: WHERE filters rows before grouping, while HAVING filters groups after aggregation. Example:

SELECT COUNT(id) FROM Orders WHERE Amount > 500 (before aggregation), vs.

SELECT COUNT(id) FROM Orders GROUP BY Status HAVING COUNT(id) > 1 (after grouping).

Question: 2c. Count total number of 'a' appearing in phrase 'Great Learning'.

Solution: SELECT LENGTH(REPLACE('Great Learning', 'a', '')) - LENGTH('Great Learning');

Counts 'a' by comparing lengths before and after removing 'a' characters.

Question: 2d. Adding and deleting a unique constraint.

Solution: ALTER TABLE tablename ADD CONSTRAINT unique_constraint_name UNIQUE(columnname);

To delete: ALTER TABLE tablename DROP CONSTRAINT unique_constraint_name;

Question: 2e. How correlated subquery can be used in Having clause?

Solution: Correlated subqueries refer to the outer query, often in the HAVING clause. Example:

SELECT department, COUNT(*) FROM employees GROUP BY department HAVING COUNT(*) > (SELECT AVG(employee_count) FROM departments);

Question: 3a. Average rainfall and evaporation per location, excluding NULL evaporation.

Solution: SELECT Location, AVG(Rainfall) AS Avg_Rainfall, AVG(Evaporation) AS

Avg_Evaporation FROM AustraliaWeather WHERE Evaporation IS NOT NULL GROUP BY Location;

Question: 3b. Max morning and afternoon temperature per location by month.

Solution: SELECT Location, MONTH(Date) AS Month, MAX(TempMorning) AS MaxMorningTemp, MAX(TempAfternoon) AS MaxAfternoonTemp FROM AustraliaWeather GROUP BY Location, Month(Date);

Question: 3c. Add Pressure9pm column with default and no null.

Solution: ALTER TABLE AustraliaWeather ADD Pressure9pm INT NOT NULL DEFAULT 1001;

Question: 3d. Compare average rainfall for Jan 2008 and July 2008.

Solution: SELECT AVG(Rainfall) AS Avg_Rainfall_Jan FROM AustraliaWeather WHERE Date BETWEEN '2008-01-01' AND '2008-01-31';

SELECT AVG(Rainfall) AS Avg_Rainfall_July FROM AustraliaWeather WHERE Date BETWEEN '2008-07-01' AND '2008-07-31';

Question: 3e. Compare total humidity for Jan and Feb 2009.

Solution: SELECT MONTH(Date) AS Month, SUM(HumidityMorning + HumidityAfternoon) AS Total_Humidity FROM AustraliaWeather WHERE YEAR(Date) = 2009 AND MONTH(Date) IN (1, 2) GROUP BY MONTH(Date);

Question: 3f. Display highest Pressure per location each month for morning weather.

Solution: SELECT Location, MONTH(Date) AS Month, MAX(PressureMorning) AS MaxPressure FROM AustraliaWeather WHERE Evaporation IS NOT NULL AND TempMorning BETWEEN 14 AND 30 AND HumidityMorning <= 70 GROUP BY Location, MONTH(Date);

Question: 4a. Query to list circuits with no races held.

Solution: SELECT CircuitID, CircuitName FROM Circuits WHERE CircuitID NOT IN (SELECT CircuitID FROM Races);

Question: 4b. Query to find racer with the shortest lap time.

Solution: SELECT DriverID, MIN(LapTime) AS ShortestTime FROM LapTimes GROUP BY DriverID ORDER BY ShortestTime LIMIT 1;

Question: 4c. Rank drivers by points accumulated.

Solution: SELECT DriverID, DriverName, SUM(Points) AS TotalPoints FROM RaceResults GROUP BY DriverID ORDER BY TotalPoints DESC;

Question: 4d. Create virtual table with race count per driver, sorted by count.

Solution: SELECT DriverID, DriverName, COUNT(RaceID) AS TotalRaces FROM RaceResults GROUP BY DriverID ORDER BY TotalRaces DESC;

Question: 4e. Report of race IDs, driver details, points, laps, and duration.

Solution: SELECT Races.RaceID, Races.RaceName, Drivers.DriverName, Results.Points, Results.Laps, Results.Duration FROM Results INNER JOIN Drivers ON Results.DriverID = Drivers.DriverID INNER JOIN Races ON Results.RaceID = Races.RaceID ORDER BY Duration;