MySQL Query Notes for Functions and Subqueries

Built-in Functions in MySQL

Character Functions

```sql
-- String concatenation
SELECT CONCAT(first_name, ' ', last_name) AS full_name FROM employees;

-- Case conversion
SELECT UPPER('hello'), LOWER('WORLD');

-- String length
SELECT LENGTH('MySQL');

-- Substring extraction
SELECT SUBSTRING('MySQL', 2, 3); -- Returns 'ySQ'

-- Trimming whitespace
SELECT TRIM('  MySQL  '), LTRIM('  MySQL'), RTRIM('MySQL  ');

-- String replacement
SELECT REPLACE('MySQL', 'SQL', 'Database');

-- Pattern matching
SELECT first_name FROM employees WHERE first_name LIKE 'J%';
```

Number Functions

```sql
-- Rounding numbers
SELECT ROUND(123.4567, 2); -- 123.46
SELECT TRUNCATE(123.4567, 2); -- 123.45

-- Absolute value
SELECT ABS(-15);

-- Ceiling and floor
SELECT CEIL(123.45), FLOOR(123.45);

-- Modulus
SELECT MOD(10, 3); -- 1
```

```sql
-- Power and square root
SELECT POWER(2, 3), SQRT(25);

-- Random number
SELECT RAND();
```

Date Functions

```sql
-- Current date and time
SELECT NOW(), CURDATE(), CURTIME();

-- Date parts extraction
SELECT YEAR(NOW()), MONTH(NOW()), DAY(NOW());
SELECT HOUR(NOW()), MINUTE(NOW()), SECOND(NOW());

-- Date arithmetic
SELECT DATE_ADD(NOW(), INTERVAL 1 DAY);
SELECT DATE_SUB(NOW(), INTERVAL 1 MONTH);
SELECT DATEDIFF('2023-12-31', '2023-01-01');

-- Date formatting
SELECT DATE_FORMAT(NOW(), '%Y-%m-%d %H:%i:%s');
SELECT DATE_FORMAT(NOW(), '%W, %M %e, %Y');

-- Day of week
SELECT DAYNAME(NOW());
```

Conversion Functions

```sql
-- Type casting
SELECT CAST('123' AS SIGNED);
SELECT CONVERT('2023-01-01', DATE);

-- Number to string
SELECT CONCAT('Order #', CAST(1001 AS CHAR));

-- Date to string
SELECT DATE_FORMAT(NOW(), '%Y-%m-%d');

-- String to date
SELECT STR_TO_DATE('January 1, 2023', '%M %d, %Y');
```

General Functions

```sql
-- Conditional logic
SELECT IF(salary > 5000, 'High', 'Low') FROM employees;

-- Case expressions
SELECT
    CASE
        WHEN salary < 3000 THEN 'Low'
        WHEN salary BETWEEN 3000 AND 6000 THEN 'Medium'
        ELSE 'High'
    END AS salary_level
FROM employees;

-- Null handling
SELECT COALESCE(commission_pct, 0) FROM employees;
SELECT IFNULL(commission_pct, 'N/A') FROM employees;

-- Greatest/least values
SELECT GREATEST(10, 20, 5), LEAST(10, 20, 5);
```

Elements of Date Format in MySQL

```sql
-- Common format specifiers:
-- %Y - 4-digit year (2023)
-- %y - 2-digit year (23)
-- %m - Month (01-12)
-- %c - Month (1-12)
-- %M - Month name (January)
-- %b - Abbreviated month (Jan)
-- %d - Day of month (01-31)
-- %e - Day of month (1-31)
-- %H - Hour (00-23)
-- %h - Hour (01-12)
-- %i - Minutes (00-59)
-- %s - Seconds (00-59)
-- %p - AM/PM
-- %W - Weekday name (Sunday)
-- %a - Abbreviated weekday (Sun)
```

Subqueries in MySQL

Subquery Uses

```sql
-- In WHERE clause
SELECT * FROM employees
WHERE department_id IN (SELECT department_id FROM departments WHERE location_
id = 1700);

-- In FROM clause (derived tables)
SELECT dept_avg.avg_salary
FROM (SELECT AVG(salary) AS avg_salary FROM employees GROUP BY department_id)
AS dept_avg
WHERE dept_avg.avg_salary > 5000;

-- In SELECT clause (scalar subqueries)
SELECT employee_id, salary, (SELECT AVG(salary) FROM employees) AS company_av
g
FROM employees;

-- In HAVING clause
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id
HAVING AVG(salary) > (SELECT AVG(salary) FROM employees);
```

Types of Subquery

Single Row Subquery

```sql
-- Returns exactly one row and one column
SELECT * FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);

-- With comparison operators (=, >, <, >=, <=, <>)
SELECT * FROM employees
WHERE job_id = (SELECT job_id FROM jobs WHERE job_title = 'Programmer');
```

Multiple Row Subquery

```sql
-- Returns multiple rows (one or more columns)
SELECT * FROM employees
WHERE department_id IN (SELECT department_id FROM departments WHERE location_
id = 1700);

-- With IN, NOT IN, ANY, ALL operators
SELECT * FROM employees
WHERE salary > ALL (SELECT salary FROM employees WHERE department_id = 90);

SELECT * FROM employees
WHERE salary > ANY (SELECT salary FROM employees WHERE department_id = 90);
```

Correlated Subquery

```sql
-- References columns from the outer query
SELECT e.first_name, e.last_name, e.salary
FROM employees e
WHERE salary > (SELECT AVG(salary) FROM employees WHERE department_id = e.dep
artment_id);

-- EXISTS/NOT EXISTS
SELECT d.department_name
FROM departments d
WHERE EXISTS (SELECT 1 FROM employees e WHERE e.department_id = d.department_
id);
```

Nested Subquery

```sql
-- Subquery within a subquery
SELECT * FROM employees
WHERE department_id IN (
    SELECT department_id FROM departments
    WHERE location_id IN (
        SELECT location_id FROM locations WHERE country_id = 'US'
    )
);
```