**SQL: Database Foundations: Sections Sections 2, 3: Select, Where, Order by, Functions**

**1-1**

Learned more about the Oracle’s APEX application, which allows to speed up the application development process. There are two parts to it: SQL Workshop and Application Builder. The basic SELECT statement is introduced along with the addition of a condition WHERE. Spelling is also important when writing out SQL commands because it can lead to errors.

**1-2**

Relational databases allow tables to be related by a common field. RDBMs are important in data storage and organizes information. Key terms for include table, column, row, field, primary key, and foreign key, and there are 6 properties for relational databases. There are also 4 categories of SQL statements: DML, DDL, TCL, DCL.

**1-3**

|  |  |
| --- | --- |
| join | Display data from two or more related tables. |
| Arithmetic operator | A symbol used to perform an operation on some values |
| column | An implementation of an attribute or relationship in a table. |
| projection | The capability in SQL to choose the columns in a table that you  want returned from a query. |
| NULL | A value that is unavailable, unassigned, unknown, or  inapplicable |
| alias | Renames a column heading. |
| Arithmetic expression | A mathematical equation |
| selection | The capability in SQL to choose the rows in a table returned from  a query |
| SELECT | Retrieves information from the database |
| SELECT | Specifies the columns to be displayed |
| FROM | Specifies the table containing the column listed in the select  clause |
| keyword | An individual SQL command |
| Clause | Part of a SQL statement |
| Statement | A combination of the two clauses |

1. SELECT first\_name, last\_name, email

FROM employees

1. SELECT last\_name AS Client, email AS "Email Address"

FROM d\_client;

1. SELECT last\_name, (salary + .50) \*.05

FROM f\_staffs;

1. C
2. b
3. Keyword: SELECT

Clause: SELECT employee\_id, last name

Statement: SELECT employee\_id, last name FROM employees

1. D
   1. Selection
   2. Projection
   3. false
   4. false
   5. true
   6. false
2. d
3. d

**2-1**

|  |  |
| --- | --- |
| DISTINCT | A command that suppresses duplicates |
| Concatenation | Links two columns together to form one character data column |
| Literal values | A group of character data |
| DESCRIBE | An SQL plus command that displays the structure of a table |

1. SELECT last\_name, mailing\_address

FROM customers;

* 1. SELECT first\_name

FROM f\_staffs;

* 1. SELECT first\_name | | ‘ ’ | | last\_name AS "DJs on Demand Clients"

FROM d\_clients;

* 1. SELECT DISCTINCT f\_order\_lines

FROM quantity;

* 1. SELECT order\_number

FROM f\_orders;

1. SELECT CONCAT('\*\*\* ', first\_name, ' \*\*\* ', first\_name, ' \*\*\*') AS "Super Star"

FROM f\_staffs

WHERE first\_name IN ('Sue', 'Bob', 'Monique');

1. A and c
2. SELECT

last\_name AS "EMPLOYEE LAST NAME",

current\_salary AS "CURRENT SALARY",

current\_salary \* 1.05 AS "SALARY WITH 5% RAISE"

FROM f\_staffs;

1. DESCRIBE EMPLOYEES;
2. SELECT

Inventory\_item AS "Inventory Item",

Cd\_title AS “CD Title”,

producer AS "Music Producer",

yr\_purchased AS “Year Purchase”

FROM D\_CDs;

1. true
2. true
3. there should be a , after last\_name. multiplication is denoted by \* not x. there is no AS for the alias. Sal is probably the wrong column name.
4. multiplication
5. d
6. projection
7. c
8. b
9. b
10. all of them
11. a
12. a

**2-2**

|  |  |
| --- | --- |
| WHERE | Restricts the rows returned by a select statement |
| Comparison operator | Compares one expression to another value or expression |

1. SELECT first\_name, last\_name, address

FROM customers

WHERE customer\_id = 456;

1. SELECT item\_name, start\_date, end\_date

FROM promotions

WHERE item\_name = 'ballpen and highlighter';

1. SELECT 'The 1997 recording in our database is ' || title AS "Oldest"

FROM recordings

WHERE year = 1997;

1. SELECT produce, title

FROM d\_cds

WHERE title = 'Carpe Diem' ;

1. SELECT cd\_title, cd\_year

FROM d\_cds

WHERE cd\_year < 2000;

1. A and b
2. SELECT studentno, fname, lname

FROM students

WHERE sex = ‘F’;

1. SELECT studentno AS “Student Number”

FROM students

WHERE major = ‘PE’;

1. SELECT \*

FROM students

WHERE sex = ‘M’;

1. SELECT titles, years

FROM dj\_table

WHERE year <> 2000

1. SELECT employee

FROM employees

WHERE birth\_date < 1980;

**2-3**

|  |  |
| --- | --- |
| ESCAPE | This option identifies that the escape characters should be  interpreted literally |
| IS NULL | Condition tests for null values |
| BETWEEN…AND | Displays rows based on a range of values |
| BETWEEN…AND | Including the specified limits and the area between them; the  numbers 1-10, inclusive |
| LIKE | Selects rows that match a character pattern |
| IN | Tests for values in a specified list of values |

1. SELECT first\_name, last\_name, salary

FROM salary

WHERE salary BETWEEN 5.00 AND 10.00;

1. SELECT location\_type, comments

FROM venues

WHERE location\_type = 'Private Home';

1. SELECT first\_name, last\_name

FROM f\_staffs

WHERE salary >= 20.00 OR salary <= 60.00;

1. SELECT title

FROM d\_cds

WHERE title LIKE '\_a%';

1. -
2. SELECT last\_name AS “Possible Candidates”

FROM employees

WHERE last\_name ‘%s’;

1. A and d
2. SELECT \*

FROM songs

WHERE type\_code IN (77, 12, 1);

**3-1**

|  |  |
| --- | --- |
| NOT | Inverts the value of the condition |
| AND | Both conditions must be true for a record to be selected |
| Logical operators | Rules that determine the order in which expressions are  evaluated and calculated |
| OR | Either condition can be true for a record to be selected |

1. The difference is in the AND and OR statements.
2. SELECT last\_name

FROM employees

WHERE last\_name LIKE '%e%' AND last\_name LIKE '%i%';

1. SELECT \*

FROM employees

WHERE hourly\_rate > 6.50 AND position <> 'order taker';

1. SELECT last\_name

FROM employees

WHERE last\_name LIKE ‘D%’ AND last\_name ‘%e%’ AND last\_name ‘%a% ;

1. -
2. c
3. SELECT first\_name, last\_name

FROM employees

WHERE hire\_date > ‘May 1998’ AND hire\_date < ‘June 1999’ AND

Salary < 8000 AND last\_name LIKE ‘%en%’;

1. SELECT email\_address

FROM employees

WHERE hire\_date = 1996 AND salary > 9000

**3-2**

|  |  |
| --- | --- |
| ORDER BY | Orders the rows in ascending order (the default order); A-Z |
| ORDER BY DESC | Orders the rows in descending order: Z-A |
| Sorting | To arrange according to class, kind, or size |

1. SELECT employee\_id AS “Number”, first\_name, last\_name

FROM employees

ORDER BY “Number”;

1. SELECT title, year

FROM d\_cds

ORDER BY year, title;

1. SELECT title AS “Our Collection”

FROM d\_songs

ORDER BY “Our Collection” DESC;

1. SELECT first\_name, last\_name, student\_id, parking\_number

FROM students\_first\_yr

WHERE year = 1

ORDER BY

last\_name ASC,

first\_name DESC;

1. SELECT department\_id, last\_name, manager\_id

FROM employees

WHERE employee\_id < 125

ORDER BY department\_id, last\_name;

Extension activities

1. d
2. c
3. a
4. a, b, c
5. c
6. d
7. b, d
8. b
9. d
10. b
11. SELECT last\_name

FROM employees

WHERE last\_name LIKE ‘St%’;

1. 0-1899 and 2101 +
   1. WHERE department\_id NOT IN (101, 102, 103);
   2. WHERE last\_name = 'King';
   3. WHERE start\_date LIKE '05-May-1998';
   4. WHERE salary BETWEEN 5000 AND 7000;
   5. WHERE id != 10;
2. 625 and 410

**3-3**

* 1. Single-row
  2. Multiple-row
  3. Single-row
  4. Multiple-row
  5. Single-row
  6. Single-row

AVG: finds the average, or mean, of a set of given values

COUNT: counts how many individual numbers are in a set of given values

MAX: finds the highest number in a set of given values

MIN: finds the lowest number in a set of given values

SUM: calculates the total of a set of given values

SELECT AVG(salary) FROM employees;

SELECT COUNT(salary) FROM employees;

SELECT MAX(salary) FROM employees;

SELECT MIN(salary) FROM employees;

SELECT SUM(salary) FROM employees;