SQL Cheat Sheet - Oracle

Create Table

Create Table:

```
CREATE TABLE tablename (
      column_name data_type
);
```

Create Table With Constraints:

```
CREATE TABLE tablename (
 column_name data_type NOT NULL,
 CONSTRAINT pkname PRIMARY KEY (col),
 CONSTRAINT fkname FOREIGN KEY (col)
      REFERENCES
      other_table(col_in_other_table),
 CONSTRAINT ucname UNIQUE (col),
 CONSTRAINT ckname CHECK (conditions)
);
```

Drop Table:

```
DROP TABLE tablename;
```

Alter Table

Add Column

```
ALTER TABLE tablename
   ADD columnname datatype;
```

Drop Column

```
ALTER TABLE tablename
   DROP COLUMN columnname;
```

Modify Column

```
ALTER TABLE tablename
   MODIFY columnname newdatatype;
```

Rename Column

```
ALTER TABLE tablename
   RENAME COLUMN currentname TO newname;
```

Add Constraint

```
ALTER TABLE tablename
 ADD CONSTRAINT constraintname
 constrainttype (columns);
```

Drop Constraint

```
ALTER TABLE tablename
 DROP CONSTRAINT constraintname;

ALTER TABLE tablename
 DROP constraint_type constraintname;
```

Rename Table

```
ALTER TABLE tablename
 RENAME TO newtablename;
```

Modifying Data

INSERT:

```
INSERT INTO tablename (col1, col2...)
 VALUES (val1, val2);
```

INSERT From Table:

```
INSERT INTO tablename (col1, col2…)
 SELECT col1, col2…
```

UPDATE:

```
UPDATE tablename
 SET col1 = val1
 WHERE condition;
```

DELETE:

```
DELETE FROM tablename
WHERE condition;
```

Common Data Types

`VARCHAR2(size)`
Text container with variable length (1 to 4000)

`CHAR(size)`
Text container with fixed length (1 to 2000)

`NUMBER(p,s)`
Numbers with precision p and scale s

`DATE`
Date preserve (1.1.-4712 to 31.12.4712)
Always contains the time to the second

SQL Cheat Sheet - Oracle

SELECT

```
SELECT col1, col2

FROM tables

WHERE conditions

GROUP BY cols

HAVING conditions

ORDER BY col1 asc, col2 desc;
```

SELECT Keywords

DISTINCT
Removes duplicate results

BETWEEN
Matches a value between two other values (inclusive)

IN
Matches a value to one of many values

NOT IN
Does not match any of many values

IS NULL
Has no value

IS NOT NULL
Has a value

LIKE
Performs partial/wildcard matches
Wildcards: **%_**

Joins

```
SELECT t1.*, t2.*
FROM t1 join_type t2
          ON t1.col = t2.col;
```

```
SELECT t1.*, t2.*
FROM t1 join_type t2 using (col)
```

[INNER] JOIN
show all matching records in both tables.

LEFT JOIN
show all records from left table, and any matching records from right table.

RIGHT JOIN
show all records from right table, and any matching records from left table.

FULL JOIN
show all records from both tables, whether there is a match or not.

CROSS JOIN
show all combinations of records from both tables.

SELF JOIN
join a table to itself. Used for hierarchical data.

```
SELECT p.*, c.*
FROM yourtable p JOIN yourtable c
                 ON p.id = c.parent_id;
```

Subqueries

```
INSERT INTO staff
       (SELECT * FROM persons);
```

```
DELETE FROM students
WHERE matriculation_number IN
  (SELECT matriculation_number
   FROM grades
   WHERE diploma_examination < 5);
```

```
UPDATE employees
   SET salary = salary * 1.2
   WHERE salary < (SELECT avg(salary)
                  FROM employees);
```

```
SELECT * FROM album_sale
   WHERE album_total > ALL (SELECT cost
                FROM album_production);
```

```
SELECT salary,
       (SELECT max(salary)
        FROM employees) as maximum
FROM employees;
```

```
SELECT first_name, last_name, dep_name
FROM (SELECT * FROM employees
       JOIN departments ON
       employees.dep_ID=departments.ID)
WHERE sex = 'f';
```

SQL Cheat Sheet - Oracle

Compound Queries

Set Operators

`UNION`
Shows unique rows from two result sets.

`UNION ALL`
Shows all rows from two result sets.

`INTERSECT`
Shows rows that exist in both result sets.

`MINUS`
Shows rows that exist in the first result set but not the second.

Aggregate Functions

`SUM()`
Finds a total of the numbers provided

`COUNT()`
Finds the number of records

`AVG()`
Finds the average of the numbers provided

`MIN()`
Finds the lowest of the numbers provided

`MAX()`
Finds the highest of the numbers provided

Common Functions

`LENGTH(string)`
Returns the length of the provided string

`INSTR(string, substring, [start_position], [occurrence])`
Returns the position of the substring within the specified string.

`SUBSTR(string, start_position, [length])`
Returns part of a value, based on a position and length.

`CEIL(input_val)`
Returns the smallest integer greater than the provided number.

`FLOOR(input_val)`
Returns the largest integer less than the provided number.

`ROUND(input_val, round_to)`
Rounds a number to a specified number of decimal places.

`TRUNC(input_value, dec_or_fmt)`
Truncates a number or date to a number of decimals or format.

`TO_CHAR(input_value, [fmt_mask])`
Converts a date or a number to a string

`TO_DATE(charvalue, [fmt_mask])`
Converts a string to a date value.

`SYSDATE`
Returns the current date, including time.

`ADD_MONTHS(input_date, num_months)`
Adds a number of months to a specified date.

`MONTHS_BETWEEN(date_1, date_2)`
returns the number of months between date_1 and date_2

Common Format Masks

YYYY: 4 digit year

YY: 2 digit year

MM: Month (01 to 12)

MON: Abbreviated month name

D: Day of week (1 to 7)

DAY: Name of day

DD: Day of month (01 to 31)

DY: Abbreviated day name

HH: Hour of day (01 to 12)

HH24: Hour of day (01 to 24)

MI: Minute of hour (00 to 59)

SS: Second of minute (00 to 59)