LIMIT / FETCH / TOP

ORACLE

```
SELECT
            COLUMN(S)
FROM
            TABLE
FETCH FIRST NUMBER ROWS ONLY;
```

MYSQL

```
SELECT
            COLUMN(S)
FROM
            TABLE
LIMIT
            NUMBER;
```

MS SQL SERVER

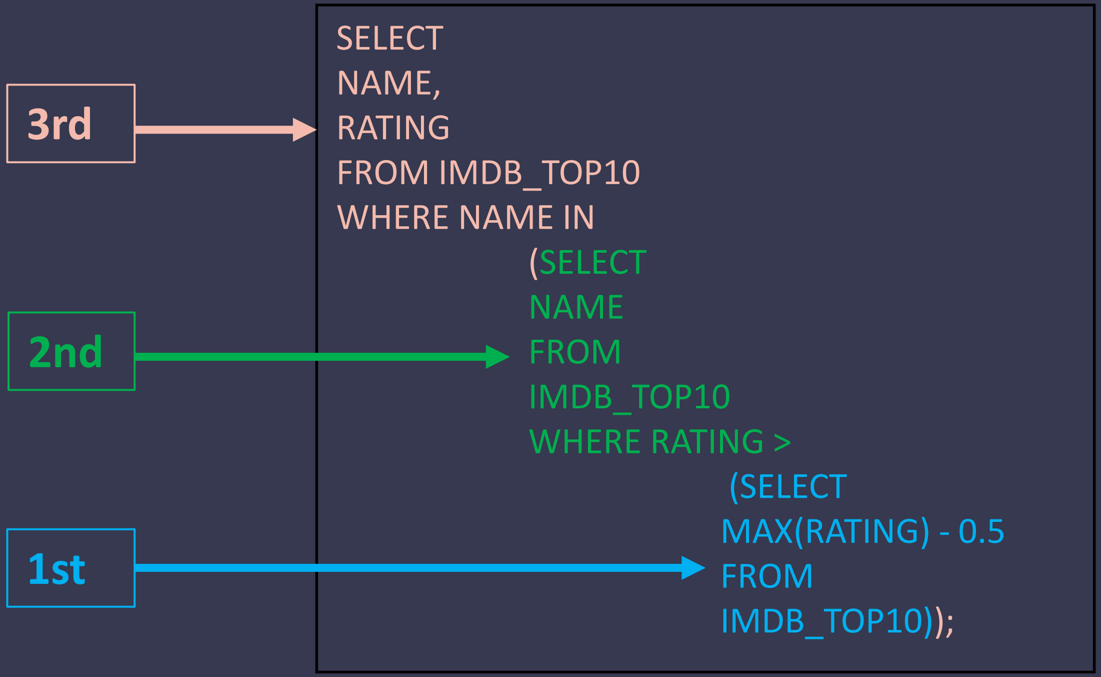
```
SELECT TOP NUMBER | PERCENT
            COLUMN(S)
FROM
            TABLE;
```

Subqueries

- A subquery is a query that is nested inside a SELECT, INSERT, UPADTE or DELETE statement
- A subquery may occur anywhere an expression is allowed, such as:
  - A SELECT clause
  - A FROM clause
  - A WHERE clause
  - A HAVING clause
  - Etc.
- Subqueries must be enclosed in parenthesis

Subquery Order of Execution

- The inner-most query is executed first
- All sub-queries follow the SQL query order of execution



```
SELECT
NAME,
RATING
FROM IMDB_TOP10
WHERE NAME IN
                (SELECT
                NAME
                FROM
                IMDB_TOP10
                WHERE RATING >
                                 (SELECT
                                MAX(RATING) - 0.5
                                FROM
                                IMDB_TOP10));
```

Example subquery 1

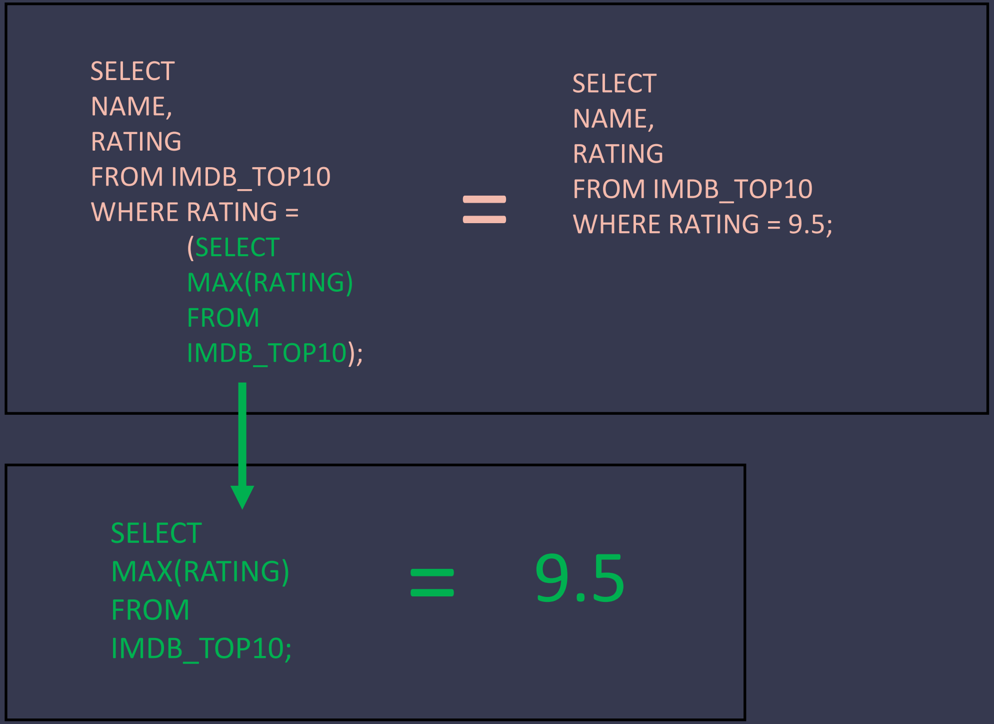
IMDB_TOP10			
NAME	**CATEGORY**	**NETWORK**	**RATING**
Breaking Bad	Drama	AMC	9.5
Game of Thrones	Adventure	HBO	9.3
The Wire	Drama	HBO	9.3
Rick and Morty	Animation	Adult Swim	9.2
Avatar: The Last Airbender	Animation	Nickelodeon	9.2
The Sopranos	Drama	HBO	9.2
Sherlock	Drama	BBC	9.1
Fullmetal Alchemist: Brotherhood	Animation	JNN	9.1
Death Note	Animation	Nippon TV	9.0
Firefly	Sci-Fi	Fox	9.0

```
SELECT
        NAME,
        RATING
FROM
    (SELECT
                NAME,
                CATEGORY,
                NETWORK
                RATING
        FROM
        IMDB_TOP10);
```

NAME	RATING
Breaking Bad	9.5
Game of Thrones	9.3
The Wire	9.3
Rick and Morty	9.2
Avatar: The Last Airbender	9.2
The Sopranos	9.2
Sherlock	9.1
Fullmetal Alchemist: Brotherhood	9.1
Death Note	9.0
Firefly	9.0

Example subquery 2

IMDB_TOP10			
NAME	CATEGORY	NETWORK	RATING
Breaking Bad	Drama	AMC	9.5
Game of Thrones	Adventure	HBO	9.3
The Wire	Drama	HBO	9.3
Rick and Morty	Animation	Adult Swim	9.2
Avatar: The Last Airbender	Animation	Nickelodeon	9.2
The Sopranos	Drama	HBO	9.2
Sherlock	Drama	BBC	9.1
Fullmetal Alchemist: Brotherhood	Animation	JNN	9.1
Death Note	Animation	Nippon TV	9.0
Firefly	Sci-Fi	Fox	9.0



```
SELECT
NAME,
RATING
FROM IMDB_TOP10
WHERE RATING =
          (SELECT
          MAX(RATING)
          FROM
          IMDB_TOP10);
```

=

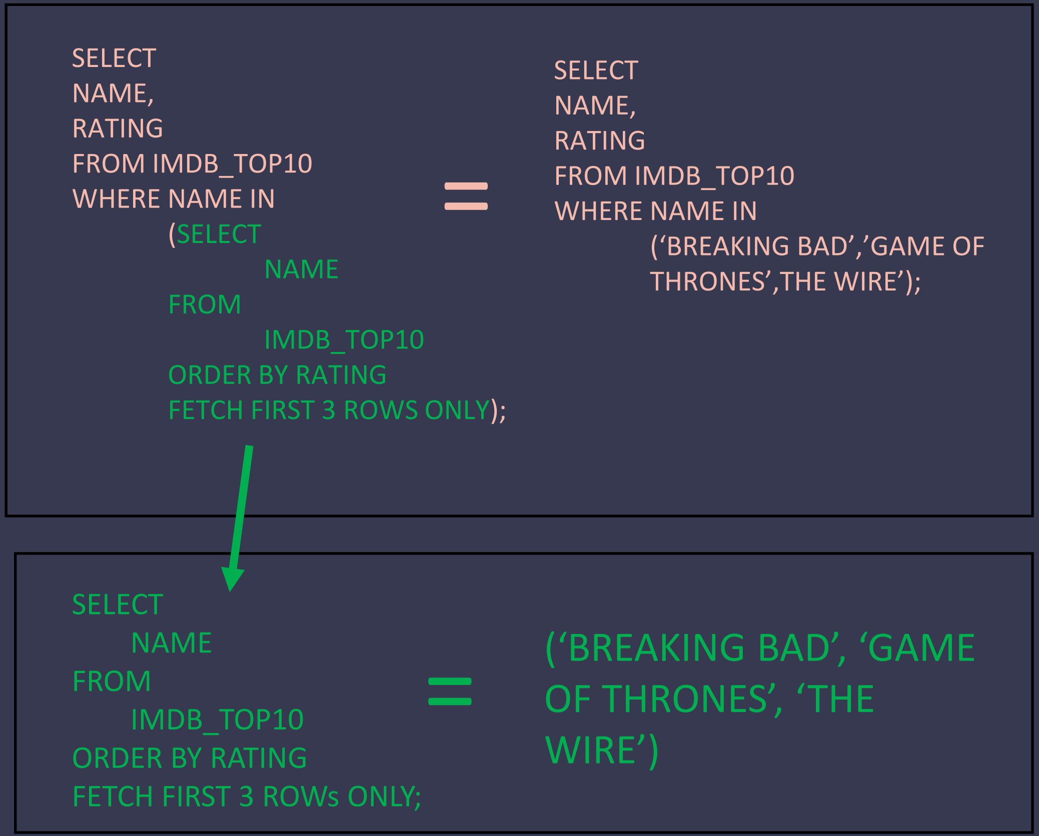
```
SELECT
NAME,
RATING
FROM IMDB_TOP10
WHERE RATING = 9.5;
```

```
SELECT
MAX(RATING)
FROM
IMDB_TOP10;
```

= 9.5

Example subquery 3

IMDB_TOP10			
NAME	CATEGORY	NETWORK	RATING
Breaking Bad	Drama	AMC	9.5
Game of Thrones	Adventure	HBO	9.3
The Wire	Drama	HBO	9.3
Rick and Morty	Animation	Adult Swim	9.2
Avatar: The Last Airbender	Animation	Nickelodeon	9.2
The Sopranos	Drama	HBO	9.2
Sherlock	Drama	BBC	9.1
Fullmetal Alchemist: Brotherhood	Animation	JNN	9.1
Death Note	Animation	Nippon TV	9.0
Firefly	Sci-Fi	Fox	9.0



```
SELECT
NAME,
RATING
FROM IMDB_TOP10
WHERE NAME IN
          (SELECT
                    NAME
          FROM
                    IMDB_TOP10
          ORDER BY RATING
          FETCH FIRST 3 ROWS ONLY);
```

=

```
SELECT
NAME,
RATING
FROM IMDB_TOP10
WHERE NAME IN
          (‘BREAKING BAD’,’GAME OF
          THRONES’,THE WIRE’);
```

```
SELECT
      NAME
FROM
      IMDB_TOP10
ORDER BY RATING
FETCH FIRST 3 ROWs ONLY;
```

=

(‘BREAKING BAD’, ‘GAME OF THRONES’, ‘THE WIRE’)

SQL Query Order Of Execution

1. FROM
2. JOIN
3. WHERE
4. GROUP BY
5. HAVING
6. SELECT
7. ORDER BY
8. LIMIT / FETCH / TOP

Joins and subqueries

How can you add a field on the IMDB_TOP10 Table that shows the average total rating across each category?

1. Group the IMDB_TOP10 table by category and select the category, avg(rating) fields

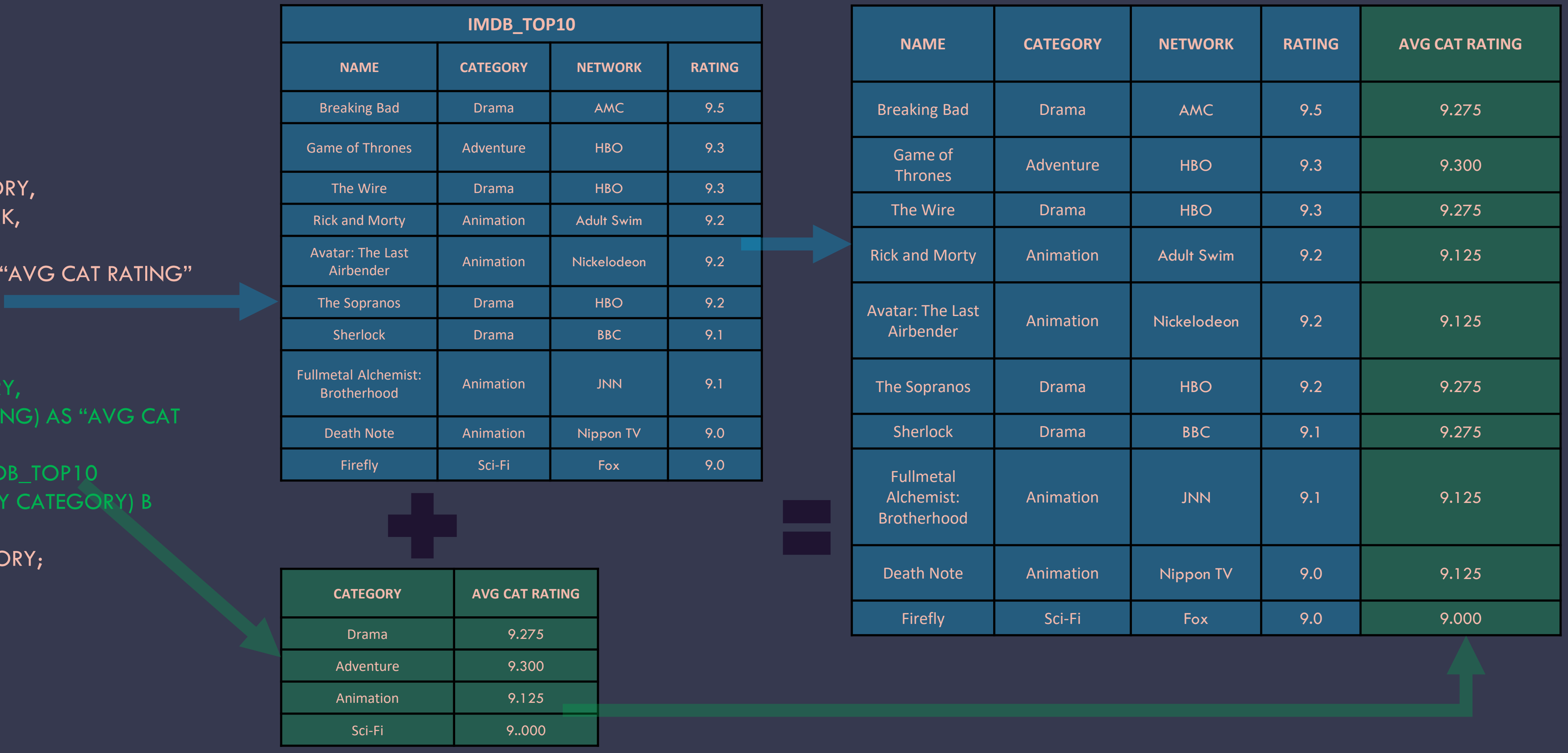
2. Join the table from step 1 to the IMDB_TOP10 table on the category field

IMDB_TOP10

NAME	CATEGORY	NETWORK	RATING	AVG CATEGORY RATING
Breaking Bad	Drama	AMC	9.5	9.275
Game of Thrones	Adventure	HBO	9.3	9.300
The Wire	Drama	HBO	9.3	9.275
Rick and Morty	Animation	Adult Swim	9.2	9.125
Avatar: The Last Airbender	Animation	Nickelodeon	9.2	9.125
The Sopranos	Drama	HBO	9.2	9.275
Sherlock	Drama	BBC	9.1	9.275
Fullmetal Alchemist: Brotherhood	Animation	JNN	9.1	9.125
Death Note	Animation	Nippon TV	9.0	9.125
Firefly	Sci-Fi	Fox	9.0	9.000

Joins and subqueries

```
SELECT
A.NAME AS NAME,
A.CATEGORY AS CATEGORY,
A.NETWORK AS NETWORK,
A.RATING As RATING,
B. “AVG CAT RATING” as “AVG CAT RATING”
FROM IMDB_TOP10 A
LEFT JOIN
                (SELECT
                CATEGORY,
                AVG(RATING) AS “AVG CAT
                RATING”
                FROM IMDB_TOP10
                GROUP BY CATEGORY) B
ON
A.CATEGORY = B.CATEGORY;
```



IMDB_TOP10			
NAME	CATEGORY	NETWORK	RATING
Breaking Bad	Drama	AMC	9.5
Game of Thrones	Adventure	HBO	9.3
The Wire	Drama	HBO	9.3
Rick and Morty	Animation	Adult Swim	9.2
Avatar: The Last Airbender	Animation	Nickelodeon	9.2
The Sopranos	Drama	HBO	9.2
Sherlock	Drama	BBC	9.1
Fullmetal Alchemist: Brotherhood	Animation	JNN	9.1
Death Note	Animation	Nippon TV	9.0
Firefly	Sci-Fi	Fox	9.0

+

CATEGORY	AVG CAT RATING
Drama	9.275
Adventure	9.300
Animation	9.125
Sci-Fi	9..000

=

NAME	CATEGORY	NETWORK	RATING	AVG CAT RATING
Breaking Bad	Drama	AMC	9.5	9.275
Game of Thrones	Adventure	HBO	9.3	9.300
The Wire	Drama	HBO	9.3	9.275
Rick and Morty	Animation	Adult Swim	9.2	9.125
Avatar: The Last Airbender	Animation	Nickelodeon	9.2	9.125
The Sopranos	Drama	HBO	9.2	9.275
Sherlock	Drama	BBC	9.1	9.275
Fullmetal Alchemist: Brotherhood	Animation	JNN	9.1	9.125
Death Note	Animation	Nippon TV	9.0	9.125
Firefly	Sci-Fi	Fox	9.0	9.000

Note on aliasing

You are not allowed to qualify functions

```
SELECT
A.NAME AS NAME,
A.CATEGORY AS CATEGORY,
A.NETWORK AS NETWORK,
A.RATING As RATING,
B. "AVG CAT RATING"
FROM IMDB_TOP10 A
LEFT JOIN
            (SELECT
            CATEGORY,
            AVG(RATING) AS "AVG CAT RATING"
            FROM IMDB_TOP10
            GROUP BY CATEGORY) B
ON
A.CATEGORY = B.CATEGORY;
```

```
SELECT
A.NAME AS NAME,
A.CATEGORY AS CATEGORY,
A.NETWORK AS NETWORK,
A.RATING As RATING,
B. AVG(RATING)
FROM IMDB_TOP10 A
LEFT JOIN
            (SELECT
            CATEGORY,
            AVG(RATING)
            FROM IMDB_TOP10
            GROUP BY CATEGORY) B
ON
A.CATEGORY = B.CATEGORY;
```