SQL ESSENTIALS FOR DATA ANALYTICS

CHRIS FRENCH

A SQL GUIDE FOR DATA PROFESSIONALS

SELECT/FROM

Query:

```
select
        full_name
from
        customers
```

Result:

full_name
Alica Reary
Delmor Rubin
Joanie Hoyt

SELECT is used to retrieve specific data or columns from a database table.

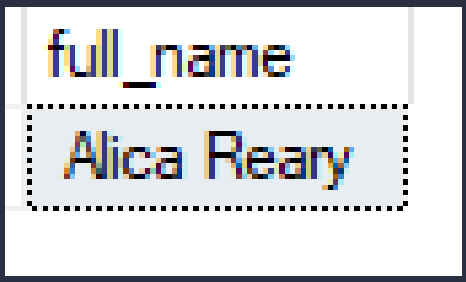
FROM is used to specify the source table from which the data should be selected in a query.

WHERE

Query:

```
select
        full_name
from
        customers
where
        full_name = 'Alica Reary'
```

Result:



WHERE allows you to filter data based on specific conditions, such as selecting rows where a certain column meets a particular criteria

GROUP BY

Query:

```
select
        sport
    ,   sum(revenue) as revenue
from
        orders
group by
        sport
```

Result:

sport	revenue
Football	94768.3199999999
Soccer	90157.93
Basketball	92116.23
Hockey	87011.9
Baseball	95363.9899999998

GROUP BY is used to group rows with similar values in one or more columns together, often in combination with aggregate functions like SUM, AVG, COUNT, etc

ORDER BY

Query:

```
select
        sport
    ,   sum(revenue) as revenue
from
        orders
group by
        sport
order by
        revenue desc
```

Result:

sport	revenue
Baseball	95363.9899999998
Football	94768.3199999999
Basketball	92116.23
Soccer	90157.93
Hockey	87011.9

ORDER BY sorts the result set by one or more columns in ascending or descending order

HAVING

Query:

```
select
        sport
    ,   sum(revenue) as revenue
from
        orders
group by
        sport
having
        sum(revenue) > 90000
order by
        revenue desc
```

Result:

sport	revenue
Baseball	95363.9899999998
Football	94768.3199999999
Basketball	92116.23
Soccer	90157.93

HAVING works with GROUP BY to filter grouped data based on conditions, similar to WHERE but for grouped data

AGGREGATE FUNCTIONS

Query:

```
select
        sport
    ,   count(*) as total_orders
    ,   sum(revenue) as revenue
    ,   avg(rating) as avg_rating
    ,   max(revenue) as top_order
    ,   min(revenue) as lowest_order

from
        orders
group by
        sport
```

Result:

sport	total_orders	revenue	avg_rating	top_order	lowest_order
Football	572	94768.3199999999	3.18987341772152	474.88	6.4
Soccer	561	90157.93	3.08	460.65	5.3
Basketball	577	92116.23	3.11111111111111	463.68	7.68
Hockey	572	87011.9	3.16379310344828	464.8	4.61
Baseball	565	95363.9899999998	3.10546875	460.32	4.61

Aggregate functions allow you to perform calculations on groups of data, such as finding the sum, average, count, minimum, or maximum value of a column

DISTINCT

Query:

```
select
    DISTINCT sport as types_of_sports
from orders
```

Result:

DISTINCT removes duplicate rows from the result set, showing only unique values.

BETWEEN/AND

Query:

```
select
        date
    ,   order_id
    ,   revenue
from
        orders
where
        date between '2022-01-01' and '2022-01-31'
```

Result:

2022-01-01 00:00:00.000	10010	178.28
2022-01-02 00:00:00.000	10011	20.94
2022-01-02 00:00:00.000	10012	134.35
2022-01-02 00:00:00.000	10013	173.17

BETWEEN filters data within a range of values.

AND is used to combine multiple conditions in a WHERE clause to filter rows from a table

LIKE/IN

Query:

```
select
        full_name
    ,   state
from
        customers
where
        full_name LIKE 'J%'
    and state IN ('California', 'New York')
```

Result:

full_name	state
Jorey Dore	New York
Joly Strase	New York
Jeffry MacCardle	New York
Jacynth Byrde	California

LIKE is used for pattern matching in text data, often with wildcard characters.

IN specifies a list of values to check against in a WHERE clause.

NULL

Query:

```
select
        order_id
    ,   rating
from
        orders
where
        rating IS NULL
```

Result:

order_id	rating
10001	NULL
10002	NULL
10003	NULL
10004	NULL
10005	NULL
10008	NULL

NULL is used to filter or check for null values in columns.

LIMIT/TOP

Query:

```
select
        top 10 order_id
from
        orders
order by
        profit desc
```

Result:

order_id
11295
11352
11369
11291
11382
10772
10898
10913
11331
10806

LIMIT/TOP limits the number of rows returned by a query, useful for sampling data or controlling the size of the result set.