SQL on 1 table

SQL is a...

- Data Definition Language (DDL)
  - Define relational schemata
  - Create/alter/delete tables and their attributes
  - We do not care
- Data Manipulation Language (DML)
  - Insert/delete/modify tuples in tables
  - Query one or more tables – discussed next!

Authors table

id	last_name	first_name	DoB	Income	Genre
1	Lopez Baranda	Christina	15/11/2000	55000	Fantasy
2	Jin-Soon	Sin	29/03/1983	65000	Crime
3	Jones	Hannah	01/02/1973	129000	Fantasy
4	Novak	Stanislaw	12/12/1992	91000	Crime
5	Turay	Tandice	09/07/1980	99000	Romance
6	Roy	Shanta	11/10/1977	55000	Fantasy
7	Berger	Henry	15/08/1956	63000	Romance
8	Khatami	Paree	11/10/1966	86000	Sci-Fi

Querying tables using SQL

```
SELECT *
FROM Authors
ORDER by last_name;
```

id	last_name	first_name	DoB	Income	Genre
7	Berger	Henry	15/08/1956	63000	Romance
2	Jin-Soon	Sin	29/03/1983	65000	Crime
3	Jones	Hannah	01/02/1973	129000	Fantasy
8	Khatami	Paree	11/10/1966	86000	Sci-Fi
1	Lopez Baranda	Christina	15/11/2000	55000	Fantasy
4	Novak	Stanislaw	12/12/1992	91000	Crime
6	Roy	Shanta	11/10/1977	55000	Fantasy
5	Turay	Tandice	09/07/1980	99000	Romance

Select From Where

Basic form

SELECT <attributes>
FROM <one or more relations>
WHERE <conditions>

The WHERE clause is optional

Selecting

Choosing some rows

```
SELECT *
FROM Authors
WHERE income >= 65000;
```

id	last_name	first_name	DoB	Income	Genre
2	Jin-Soon	Sin	29/03/1983	65000	Crime
3	Jones	Hannah	01/02/1973	129000	Fantasy
4	Novak	Stanislaw	12/12/1992	91000	Crime
5	Turay	Tandice	09/07/1980	99000	Romance
8	Khatami	Paree	11/10/1966	86000	Sci-Fi

Selecting 2

```
SELECT *
FROM Authors
WHERE last_name = 'Jones';
```

id	last_name	first_name	DoB	Income	Genre
3	Jones	Hannah	01/02/1973	129000	Fantasy

The result is a table

Projection

Choosing some columns

```
SELECT last_name, first_name, income
FROM Authors
WHERE income >= 65000;
```

last_name	first_name	Income
Jin-Soon	Sin	65000
Jones	Hannah	129000
Novak	Stanislaw	91000
Turay	Tandice	99000
Khatami	Paree	86000

Projection

Selection before projection

```
SELECT last_name, first_name
FROM Authors
WHERE income >= 65000;
```

last_name	first_name
Jin-Soon	Sin
Jones	Hannah
Novak	Stanislaw
Turay	Tandice
Khatami	Paree

Remember

- commands are case **insensitive**
- value are case **sensitive**
- single quotes 'Jones', not "Jones"

Duplicates

```
SELECT Genre
FROM Authors;
```

Genre
Fantasy
Crime
Fantasy
Crime
Romance
Fantasy
Romance
Sci-Fi

Duplicates 2

```
SELECT DISTINCT Genre
FROM Authors;
```

Genre
Fantasy
Crime
Romance
Sci-Fi

Compound conditions

```
SELECT *
FROM Authors
WHERE income >= 65000 AND Genre = 'Crime';
```

id	last_name	first_name	DoB	Income	Genre
2	Jin-Soon	Sin	29/03/1983	65000	Crime
4	Novak	Stanislaw	12/12/1992	91000	Crime

Conditions on strings

- % any string
- _ any character
- use **LIKE**

Find all authors whose first name begins with “S”

```
SELECT *
FROM Authors
WHERE first_name LIKE 'S%';
```

id	last_name	first_name	DoB	Income	Genre
2	Jin-Soon	Sin	29/03/1983	65000	Crime
4	Novak	Stanislaw	12/12/1992	91000	Crime
6	Roy	Shanta	11/10/1977	55000	Fantasy

Conditions on strings

Find all authors whose first name begins with "*S*" and the third letter is "*a*"

```
SELECT *
FROM Authors
WHERE first_name LIKE 'S_a%';
```

id	last_name	first_name	DoB	Income	Genre
4	Novak	Stanislaw	12/12/1992	91000	Crime
6	Roy	Shanta	11/10/1977	55000	Fantasy

Ordering

```
SELECT *
FROM Authors
WHERE income >= 65000
ORDER BY last_name;
```

id	last_name	first_name	DoB	Income	Genre
2	Jin-Soon	Sin	29/03/1983	65000	Crime
3	Jones	Hannah	01/02/1973	129000	Fantasy
8	Khatami	Paree	11/10/1966	86000	Sci-Fi
4	Novak	Stanislaw	12/12/1992	91000	Crime
5	Turay	Tandice	09/07/1980	99000	Romance

Ordering 2

```
SELECT *
FROM Authors
WHERE income >= 65000
ORDER BY Genre, DESC income;
```

id	last_name	first_name	DoB	Income	Genre
4	Novak	Stanislaw	12/12/1992	91000	Crime
2	Jin-Soon	Sin	29/03/1983	65000	Crime
3	Jones	Hannah	01/02/1973	129000	Fantasy
5	Turay	Tandice	09/07/1980	99000	Romance
8	Khatami	Paree	11/10/1966	86000	Sci-Fi

Count

```
SELECT count(*)
FROM Authors
WHERE income >= 65000;
```

count
5

Counts the number of rows

Count 2

```
SELECT count(DoB, first_name)
FROM Authors
WHERE income >= 65000;
```

DoB	first_name
5	5

Counts the number of rows without null values

AVG

```
SELECT AVG(income)
FROM Authors
WHERE income >= 65000;
```

avg
94000

Also SUM, MIN, MAX

Group By

For each genre, count the number of authors with income larger than 65000

```
SELECT genre, count(*)
FROM Authors
WHERE income >= 65000
GROUP BY genre;
```

genre	count(*)
Fantasy	1
Crime	2
Romance	1
Sci-Fi	1

Group By

For each genre, find the average income

```
SELECT genre, avg(income)
FROM Authors
GROUP BY genre;
```

genre	avg(income)
Fantasy	79666.67
Crime	78000
Romance	81000
Sci-Fi	86000

Attributes

```
SELECT genre, last_name, avg(income)
FROM Authors
GROUP BY genre;
```

Is not valid

SELECT can have:

- attributes in GROUP BY
- aggregate operators

Having

Extract the genres that have at least two authors with income larger than 65000

```
SELECT genre, count(*) as quanti
FROM Authors
WHERE income >= 65000
GROUP BY genre
HAVING count(*) >= 2;
```

genre	quanti
Crime	2

Reading order

1. FROM
2. WHERE
3. GROUP BY
4. HAVING
5. SELECT

License