

Introduction to Databases

(INFR10080)

(Aggregation)

Instructor: Yang Cao

(Fall 2025)



THE UNIVERSITY
of EDINBURGH

Changelog

v25.0 Initial version

Duplicates and aggregation (1)

Customer			
ID	Name	City	Age
1	John	Edinburgh	31
2	Mary	London	37
3	Jane	London	22
4	Jeff	Cardiff	22

Average age of customers: $\text{avg}(\pi_{\text{Age}}(\text{Customer}))$

- If we remove duplicates we get $\frac{31+37+22}{3} = 30$ (**wrong**)

SQL keeps duplicates by default:

```
SELECT AVG(age)
FROM Customer ;
```

2/20

Duplicates and aggregation (2)

Account			
Number	Branch	CustID	Balance
111	London	1	1330.00
222	London	2	1756.00
333	Edinburgh	1	450.00

Number of branches: $|\varepsilon(\pi_{\text{Branch}}(\text{Account}))|$

- If we keep duplicates we get 3 (**wrong**)

In SQL:

```
SELECT COUNT(DISTINCT branch)
FROM Account ;
```

3/20

Aggregate functions in SQL

COUNT number of elements in a column

AVG average value of elements in a column

SUM adds up all elements in a column

MIN minimum value of elements in a column

MAX maximum value of elements in a column

- Using **DISTINCT** with **MIN** and **MAX** makes no difference
- **COUNT**(*) counts all rows in a table
- **COUNT**(**DISTINCT** *) is **illegal**

To count all **distinct** rows of a table *T* use

```
SELECT COUNT (DISTINCT T.*)  
FROM    T ;
```

4/20

Aggregation and empty tables

Suppose table *T* has a column (of numbers) called *A*

```
SELECT MIN (A) , MAX (A) , AVG (A) , SUM (A) , COUNT (A) , COUNT (*)  
FROM    T  
WHERE    1=2 ;
```

min	max	sum	avg	count	count
				0	0

5/20

Aggregation with grouping

Account			
Num	Branch	CID	Balance
111	London	1	1330.00
222	London	2	1756.00
333	Edinburgh	1	450.00

How much money does **each** customer have in total **across all of his accounts**?

Idea

1. Partition **Account** into **groups** (one per customer) of rows
2. Sum balances in each group separately
3. Take the union of the results for each group

6/20

Account			
Num	Branch	CID	Balance
111	London	1	1330.00
222	London	2	1756.00
333	Edinburgh	1	450.00

Num	Branch	CID	Balance
111	London	1	1330.00
333	Edinburgh	1	450.00

		CID	SUM
		1	1780.00

Num	Branch	CID	Balance
222	London	2	1756.00

		CID	SUM
		2	1756.00

CID	SUM
1	1780.00
2	1756.00

7/20

Grouping in SQL

Account

Number	Branch	CustID	Balance
111	London	1	1330.00
222	London	2	1756.00
333	Edinburgh	1	450.00

How much money does **each** customer have in total **across all of his accounts**?

```
SELECT    A.custid, SUM(A.balance)
FROM      Account A
GROUP BY  A.custid ;
```

Answer:

CustID	SUM
1	1780.00
2	1756.00

8/20

Account

Number	Branch	CustID	Balance
111	London	1	1330.00
222	London	2	1756.00
333	Edinburgh	1	450.00

CustID	Balance
1	1330.00
1	450.00

CustID	Balance
2	1756.00

CustID	SUM
1	1780.00

CustID	SUM
2	1756.00

CustID	SUM
1	1780.00
2	1756.00

9/20

Grouping in SQL: Another example

Account

Number	Branch	CustID	Balance
111	London	1	1330.00
222	London	2	1756.00
333	Edinburgh	1	450.00

How much money is there in **total in each branch**?

```
SELECT    A.branch, SUM(A.balance)
FROM      Account A
GROUP BY  A.branch ;
```

Answer:

Branch	SUM
London	3086.00
Edinburgh	450.00

10/20

Account

Number	Branch	CustID	Balance
111	London	1	1330.00
222	London	2	1756.00
333	Edinburgh	1	450.00

Branch	Balance
London	1330.00
London	1756.00

Branch	Balance
Edinburgh	450.00

Branch	SUM
London	3086.00

Branch	SUM
Edinburgh	450.00

Branch	SUM
London	3086.00
Edinburgh	450.00

11/20

YAGE: Yet Another Grouping Example

Account			
Number	Branch	CustID	Balance
111	London	1	1330.00
222	London	2	1756.00
333	Edinburgh	1	450.00
444	Edinburgh	2	613.00
555	London	1	300.00
666	Edinburgh	2	1217.00

How much money does **each customer** have in **each branch**?

```
SELECT  A.custid, A.branch, SUM(A.balance)
FROM    Account A
GROUP BY A.custid, A.branch ;
```

Answer:

CustID	Branch	SUM
1	London	1630.00
1	Edinburgh	450.00
2	London	1756.00
2	Edinburgh	1830.00

12/20

Account			
Number	Branch	CustID	Balance
111	London	1	1330.00
222	London	2	1756.00
333	Edinburgh	1	450.00
444	Edinburgh	2	613.00
555	London	1	300.00
666	Edinburgh	2	1217.00

Branch	CustID	Balance
London	1	1330.00
London	1	300.00

Branch	CustID	Balance
London	2	1756.00

Branch	CustID	Balance
Edinburgh	1	450.00

Branch	CustID	Balance
Edinburgh	2	613.00
Edinburgh	2	1217.00

Branch	CustID	SUM
London	1	1630.00

Branch	CustID	SUM
London	2	1756.00

Branch	CustID	SUM
Edinburgh	1	450.00

Branch	CustID	SUM
Edinburgh	2	1830.00

CustID	Branch	SUM
1	London	1630.00
1	Edinburgh	450.00
2	London	1756.00
2	Edinburgh	1830.00

13/20

Beware

Without constraints, in queries with **GROUP BY**, attributes in **SELECT** **must**

- appear in the **GROUP BY**
- or**
- be used in an aggregate function

The following query gives an error:

```
SELECT    A.custid, A.branch, SUM(A.balance)
FROM      Account A
GROUP BY  A.branch ;
```

14/20

Filtering based on aggregation

Account			
Number	Branch	CustID	Balance
111	London	1	1330.00
222	London	2	1756.00
333	Edinburgh	1	450.00

Branches with a total balance (across accounts) of at least 500?

```
SELECT    A.branch, SUM(A.balance)
FROM      Account A
GROUP BY  A.branch
HAVING    SUM(A.balance) >= 500 ;
```

Answer:	Branch	SUM
	London	3086.00

15/20

Order of evaluation

1. Take rows from the (joined) tables listed in **FROM**
2. Discard rows not satisfying the **WHERE** condition
3. Partition rows according to attributes in **GROUP BY**
4. **Compute aggregates**
5. Discard rows not satisfying the **HAVING** condition
6. Output the values of expressions listed in **SELECT**

16/20

This frame intentionally left blank.

Aggregation and arithmetic (1)

Account

Number	Branch	CustID	Balance	Spend
111	London	1	1330.00	250.00
222	London	2	1756.00	356.00
333	Edinburgh	1	450.00	0.00

Money available in total to each customer across his accounts

```
SELECT A.custid, SUM(A.balance - A.spend)
FROM Account A
GROUP BY A.custid ;
```

Answer:

CustID	SUM
1	1530.00
2	1400.00

17/20

Account

Number	Branch	CustID	Balance	Spend
111	London	1	1330.00	250.00
222	London	2	1756.00	356.00
333	Edinburgh	1	450.00	0.00

CustID	Balance	Spend	CustID	Balance	Spend
1	1330.00	250.00	2	1756.00	356.00
1	450.00	0.00			
CustID	Balance – Spend		CustID	Balance – Spend	
1	1080.00		2	1400.00	
1	450.00				
CustID	SUM		CustID	SUM	
1	1530.00		2	1400.00	
CustID	SUM		CustID	SUM	
1	1530.00		2	1400.00	

18/20

Aggregation and arithmetic (2)

Account

Number	Branch	CustID	Balance	Spend
111	London	1	1330.00	250.00
222	London	2	1756.00	356.00
333	Edinburgh	1	450.00	0.00

Money available in total to each customer across his accounts

```
SELECT  A.custid, SUM(A.balance) - SUM(A.spend)
FROM    Account A
GROUP BY A.custid ;
```

Answer:

CustID	?column?
1	1530.00
2	1400.00

19/20

Account

Number	Branch	CustID	Balance	Spend
111	London	1	1330.00	250.00
222	London	2	1756.00	356.00
333	Edinburgh	1	450.00	0.00

CustID	Balance	Spend	CustID	Balance	Spend
1	1330.00	250.00	2	1756.00	356.00
1	450.00	0.00			
CustID	SUM	SUM	CustID	SUM	SUM
1	1780.00	250.00	2	1756.00	356.00
CustID	?column?		CustID	?column?	
1	1530.00		2	1400.00	

CustID	?column?
1	1530.00
2	1400.00

20/20