sum & sum with group by

sum → find sum of all pages.

book_id	title	author_fname	author_lname	released_year	stock_quantity	pages
1	The Namesake	Jhumpa	Lahiri	2003	32	291
2	Norse Mythology	Neil	Gaiman	2016	43	304
3	American Gods	Neil	Gaiman	2001	12	465
4	Interpreter of Maladies	Jhumpa	Lahiri	1996	97	198
5	A Hologram for the King: A Novel	Dave	Eggers	2012	154	352
6	The Circle	Dave	Eggers	2013	26	504
7	The Amazing Adventures of Kavalier & Clay	Michael	Chabon	2000	68	634
8	Just Kids	Patti	Smith	2010	55	304
9	A Heartbreaking Work of Staggering Genius	Dave	Eggers	2001	104	437
10	Coraline	Neil	Gaiman	2003	100	208
11	What We Talk About When We Talk About Love: Stories	Raymond	Carver	1981	23	176
12	Where I'm Calling From: Selected Stories	Raymond	Carver	1989	12	526
13	White Noise	Don	DeLillo	1985	49	320
14	Cannery Row	John	Steinbeck	1945	95	181
15	Oblivion: Stories	David	Foster Wallace	2004	172	329
16	Consider the Lobster	David	Foster Wallace	2005	92	343
17	10% Happier	Dan	Harris	2014	29	256
18	fake_book	Freida	Harris	2001	287	428
19	Lincoln In The Bardo	George	Saunders	2017	1000	367

sum with group by

find sum of all pages each author has written.

avg and avg with group by

$$\text{avg} = \frac{\text{sum()}}{\text{count()}} = \frac{\text{sum of all values}}{\text{total no. of observations}}$$

↓

upto 4 decimal pts

avg → find avg of all the books present in store.

avg → find avg of all the books present in store.

avg with group by

find avg of all the books left for a particular book for every author.

find avg of books released in each year.

Relationships

Types of relationship (mappings)

a). One to one

country — capital

username — email

akash — pan card

b). One to many

country → cities

parent → child

manager → reportees

c). many to many

teacher ⟷ student

shopkeeper ⟷ customers

Example One to many (1:M)

Customers Order

primary key → id
fname
lname
email

(parent table)

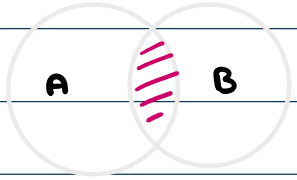
id → primary key
o_date
amount
cus_id → foreign key

(child table)

id	first_name	last_name	email
1	Boy	George	george@gmail.com
2	George	Michael	gm@gmail.com
3	David	Bowie	david@gmail.com
4	Blue	Steele	blue@gmail.com
5	Bette	Davis	bette@aol.com

id	order_date	amount	customer_id
1	2016-02-10	100	1
2	2017-11-11	36	1
3	2014-12-12	801	2
4	2015-01-03	13	2
5	1999-04-11	450	5

Inner join



Select all records from A and B where the join condition is met.

id	first_name	last_name	email
1	Boy	George	george@gmail.com
2	George	Michael	gm@gmail.com
3	David	Bowie	david@gmail.com
4	Blue	Steele	blue@gmail.com
5	Bette	Davis	bette@aol.com

id	order_date	amount	customer_id
1	2016-02-10	100	1
2	2017-11-11	36	1
3	2014-12-12	801	2
4	2015-01-03	13	2
5	1999-04-11	450	5

customer.id , customer.first_name

order.order_date , order.id

customer

id	first_name	last_name	email
1	Boy	George	george@gmail.com
2	George	Michael	gm@gmail.com
~~3~~	~~David~~	~~Bowie~~	~~david@gmail.com~~
~~4~~	~~Blue~~	~~Steele~~	~~blue@gmail.com~~
5	Bette	Davis	bette@aol.com

order

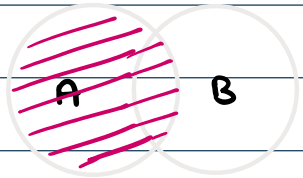
id	order_date	amount
1	2016-02-10	100
3	2014-12-12	801
5	1999-04-11	450
2	2017-11-11	30
4	2015-01-03	13

1 Boy George -

2 George Michael -

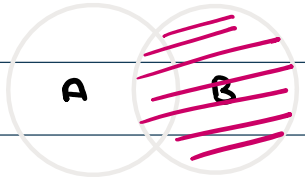
condition => customer.id = = order.customer_id

Left join



Select everything from A, along with any matching records in B.

Right join



Select everything from B, along with any matching records in A.

Data types

char v/s varchar

char	varchar
1. fixed length	1. variable length
2. faster	
3. Example :- state abbrevations - UP, HR gender → M / F phone no → 10 digits	5. allocate space according to the data size passed at the run time
4. char(5) → 5 spaces	
5. allocate space at declaration	

Decimal (M, D)