Level 2-Practice

>> المهارة مش بتتبني من الشرح...

>> بتتبني من الممارسة

>> Every query teaches something

>> Even the ones that don't work. :)

15 SQL Practice Questions & Solutions

-- Amr mousa

Practice
Reflect
Repeat

Level 2 - practice

#1 - The list of all books (book title, year of publication) published between 1997 and 2005 and ordered in ascending by the year of publication (note - there are two separate tables for books: Book1 table is for books published until 1999 and the book2 table is for books published after 1999).

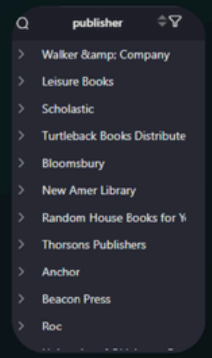
```sql
SELECT book_title, year_of_publication
FROM books_schema.books1
WHERE year_of_publication >=1997
UNION
SELECT book_title, year_of_publication
FROM books_schema.books2
WHERE year_of_publication <=2005
ORDER BY year_of_publication ASC
```

book_title	year_of_publicati integer
The Annulet of Gilt: An Ase	2005
Snobs	2005
The Cape Cod Mystery: An	2005
The Cornish Trilogy	2005
An Oblique Approach	2005
Serpico	2005
Josie and Jack : A Novel	2005
The Awful Secret : A Crowne	2005
Thunderball (James Bond 0(	2005
The Five Love Languages: Fi	2005
Me & Emma	2005

Amr Mousa

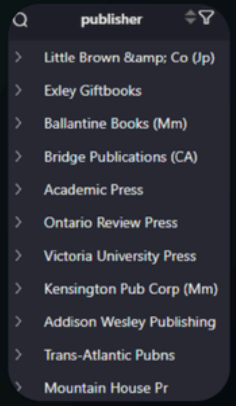
#2 - The list of publishers that published books in 1990 and 2004 (only publishers that published books in both years).

```
SELECT DISTINCT publisher  from books_schema.books1
WHERE year_of_publication = 1990
INTERSECT
SELECT DISTINCT publisher from books_schema.books2
WHERE year_of_publication =2004
```



#3 - The list of publishers that published books in 1990 but not in 2004.

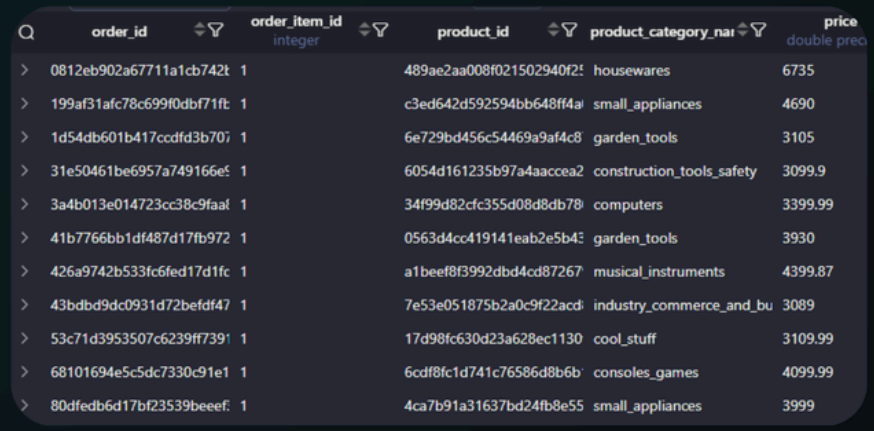
```
SELECT DISTINCT publisher  from books_schema.books1
WHERE year_of_publication = 1990
EXCEPT
SELECT DISTINCT publisher from books_schema.books2
WHERE year_of_publication =2004
```



Amr Mousa

#4 - List all order items (order id, order item id, price) with the product information (product id, product ,category name), with a price higher than 3000, order by the order id and then order item id.

```sql
SELECT
o.order_id ,
o.order_item_id ,
p.product_id ,
p.product_category_name,
o.price
FROM ecommerce_schema.order_items as o
INNER JOIN ecommerce_schema.products as p on o.product_id = p.product_id
WHERE price > 3000
order BY 1 ,2 desc;
```



order_id	order_item_id integer	product_id	product_category_nai	price double prec
0812eb902a67711a1cb742t	1	489ae2aa008f021502940f2	housewares	6735
199af31afc78c699f0dbf71ft	1	c3ed642d592594bb648ff4a	small_appliances	4690
1d54db601b417ccdfd3b707	1	6e729bd456c54469a9af4c8	garden_tools	3105
31e50461be6957a749166e9	1	6054d161235b97a4aaccea2	construction_tools_safety	3099.9
3a4b013e014723cc38c9faa	1	34f99d82cfc355d08d8db78	computers	3399.99
41b7766bb1df487d17fb972	1	0563d4cc419141eab2e5b43	garden_tools	3930
426a9742b533fc6fed17d1fc	1	a1beef8f3992dbd4cd87267	musical_instruments	4399.87
43bdbd9dc0931d72befdf47	1	7e53e051875b2a0c9f22acd	industry_commerce_and_bu	3089
53c71d3953507c6239ff7391	1	17d98fc630d23a628ec1130	cool_stuff	3109.99
68101694e5c5dc7330c91e1	1	6cdf8fc1d741c76586d8b6b	consoles_games	4099.99
80dfedb6d17bf23539beeef	1	4ca7b91a31637bd24fb8e55	small_appliances	3999

Amr Mousa

#5 - List all orders (order id, order status) that order status is “shipped” with their order items (order item id, price) with the product information (product id, product category name) with a price between 50 and 60, order by the order id and then order item id.

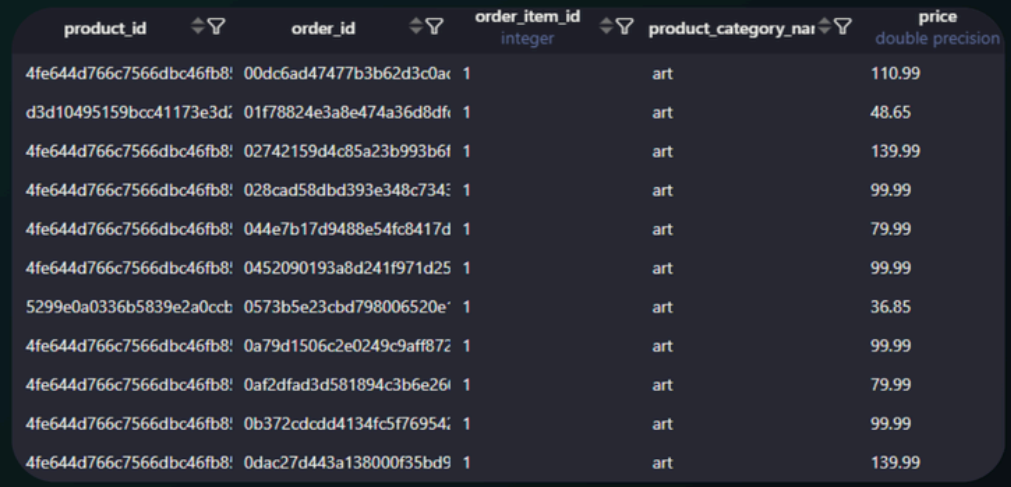
```sql
SELECT
o.order_id  ,
oi.order_item_id ,
p.product_id ,
p.product_category_name ,
oi.price ,
o.order_status
FROM ecommerce_schema.orders as o
left JOIN ecommerce_schema.order_items as oi ON oi.order_id = o.order_id
left JOIN ecommerce_schema.products as p on p.product_id = oi.product_id
WHERE o.order_status = 'shipped' AND price BETWEEN 50 AND 60
order by 1,2;
```

order_id	order_item_id (integer)	product_id	product_category_nar	price (double precision)	order_status
002f19a65a2ddd70a090297	1	9eae06d51aaa383b2bed55	bed_bath_table	53.98	shipped
00a99c50fdff7e36262caba3	1	76d75f398634bf194ba99d7	computers_accessories	52.99	shipped
08af87fbe33eeebc7840433	2	d902ca64b614964dc31d6b	furniture_decor	52.3	shipped
08af87fbe33eeebc7840433	3	d902ca64b614964dc31d6b	furniture_decor	52.3	shipped
09442375307dcece0366e8	1	7c1bd920dbdf22470b68bd	health_beauty	58.99	shipped
0cd059c0a0fa574521d2038	1	7c1bd920dbdf22470b68bd	health_beauty	58.99	shipped
11ac4935b48d92b0e8d9b2	1	fa7be99321f2dbb1e518d21	sports_leisure	59.99	shipped
170668dbc6b505ada8c063	1	232ae93f6ac6e6a57c1b34b	auto	54.9	shipped
186ac860f8928db64590e1f	1	ea44caac707f7f1325182a53	baby	52	shipped
1ae35fdd1e36d167c790179	1	4e72c18c19fee00460f9504	furniture_decor	59.88	shipped
20343beb23f60b10de8ede	1	4a9947ec1fcec2b3321193e	watches_gifts	59	shipped

Amr Mousa

#6 - List of all products (product id, product category name) from the category “art” with their order items (order id, order item id, price) whether the product was ordered or not.

```
select
p.product_id ,
o.order_id ,
o.order_item_id ,
p.product_category_name ,
o.price
FROM ecommerce_schema.products as p
left JOIN ecommerce_schema.order_items as o on p.product_id = o.product_id
WHERE p.product_category_name ='art';
```



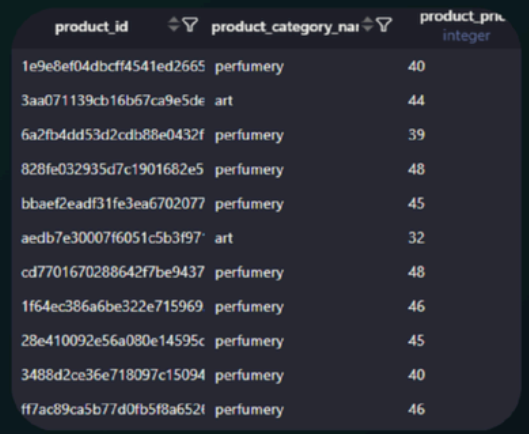
product_id	order_id	order_item_id integer	product_category_nai	price double precision
4fe644d766c7566dbc46fb8!	00dc6ad47477b3b62d3c0ac	1	art	110.99
d3d10495159bcc41173e3d:	01f78824e3a8e474a36d8df	1	art	48.65
4fe644d766c7566dbc46fb8!	02742159d4c85a23b993b6f	1	art	139.99
4fe644d766c7566dbc46fb8!	028cad58dbd393e348c734:	1	art	99.99
4fe644d766c7566dbc46fb8!	044e7b17d9488e54fc8417d	1	art	79.99
4fe644d766c7566dbc46fb8!	0452090193a8d241f971d25	1	art	99.99
5299e0a0336b5839e2a0ccb	0573b5e23cbd798006520e	1	art	36.85
4fe644d766c7566dbc46fb8!	0a79d1506c2e0249c9aff87:	1	art	99.99
4fe644d766c7566dbc46fb8!	0af2dfad3d581894c3b6e26	1	art	79.99
4fe644d766c7566dbc46fb8!	0b372cdcdd4134fc5f76954:	1	art	99.99
4fe644d766c7566dbc46fb8!	0dac27d443a138000f35bd9	1	art	139.99

Amr Mousa

Level 2 - Practice

#7 - All products from the “art” or “perfumery” category (product id, product category name, product price) that the product price is less than the average price of all products.

```sql
SELECT product_id , product_category_name ,product_price
FROM ecommerce_schema.products
WHERE product_category_name IN ('art', 'perfumery')
AND product_price < (SELECT avg(product_price)
FROM ecommerce_schema.products);
```



product_id	product_category_nar	product_pri integer
1e9e8ef04dbcff4541ed2665	perfumery	40
3aa071139cb16b67ca9e5de	art	44
6a2fb4dd53d2cdb88e0432f	perfumery	39
828fe032935d7c1901682e5	perfumery	48
bbaef2eadf31fe3ea6702077	perfumery	45
aedb7e30007f6051c5b3f97	art	32
cd77016702886427f7be9437	perfumery	48
1f64ec386a6be322e715969	perfumery	46
28e410092e56a080e14595c	perfumery	45
3488d2ce36e718097c15094	perfumery	40
ff7ac89ca5b77d0fb5f8a652(	perfumery	46

Amr Mousa

#8 - List all the names of customers that have orders with a “delivered” status with the products category “art” and price is more than 120

```sql
SELECT customer_name
FROM ecommerce_schema.customers
WHERE customer_id in
(SELECT customer_id
FROM ecommerce_schema.orders
WHERE order_status='delivered'
AND order_id IN
(SELECT order_id
FROM ecommerce_schema.order_items
WHERE price > 120
AND product_id IN
(SELECT product_id
FROM ecommerce_schema.products
WHERE product_category_name ='art')))
```

customer_name
Pierre
Stanley
Stewart
Barney
Russel
Dennis
Wash
Olan
Otho
Marshall
Elden

Amr Mousa

#9 - List all orders reviews (order id, review score) and transform the review score (scale 1 to 5) into a wording scale ('Very Dissatisfied', 'Dissatisfied', 'Neutral', 'Satisfied', 'Very Satisfied')

```sql
select  order_id,
CASE review_score
  WHEN '1' THEN 'Very Dissatisfied'
  WHEN '2' THEN 'Dissatisfied'
  WHEN '3' then 'Neutral'
  WHEN '4' THEN 'Satisfied'
  ELSE 'Very Satisfied'
END as review_score_test
FROM ecommerce_schema.order_reviews;
```

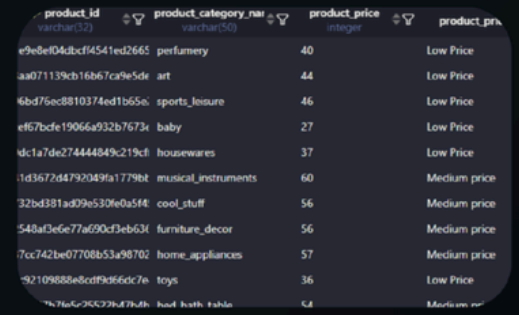
order_id varchar(32)	review_score_test
73fc7af87114b39712e6da7!	Satisfied
a548910a1c6147796b98fdf7	Very Satisfied
f9e4b658b201a9f2ecdecbb!	Very Satisfied
658677c97b385a9be170737	Very Satisfied
8e6bfb81e283fa7e4f11123a	Very Satisfied
b18dcdf73be66366873cd26	Very Dissatisfied
e48aa0d2dcec3a2e8734881	Very Satisfied
c31a859e34e3adac22f3769!	Very Satisfied
9c214ac970e84273583ab52	Very Satisfied
b9bf720beb4ab372876008!	Satisfied
cdf9aa68e72324eeb25c7de!	Very Satisfied

Amr Mousa

#10 -List all products (product id, product category, product price) and add a new column that will classify each product into a specific price category:

- Price<=50 → “Low Price”
- 50<Price<100 → “Medium Price”
- Price>100 → “High Price”

```sql
SELECT product_id ,product_category_name ,product_price
CASE
 WHEN product_price <= 50 THEN 'Low Price'
 WHEN product_price > 50 and product_price < 100 THEN 'Medium price'
 WHEN product_price >100 THEN 'High Price'
 ELSE 'others'
END as product_price_j
FROM ecommerce_schema.products;
```



product_id varchar(32)	product_category_nar varchar(50)	product_price integer	product_pri
e9e8ef04dbcff4541ed2665	perfumery	40	Low Price
aa071139cb16b67ca9e5de	art	44	Low Price
6bd76ec8810374ed1b65e	sports_leisure	46	Low Price
ef67bcfe19066a932b7673	baby	27	Low Price
dc1a7de274444849c219cf	housewares	37	Low Price
1d3672d4792049fa1779bb	musical_instruments	60	Medium price
32bd381ad09e530fe0a5f4	cool_stuff	56	Medium price
548af3e6e77a690cf3eb63	furniture_decor	56	Medium price
7cc742be07708b53a98702	home_appliances	57	Medium price
92109888e8cdf9d66dc7e	toys	36	Low Price
	bed_bath_table	54	

Amr Mousa

Level 2 - Practice

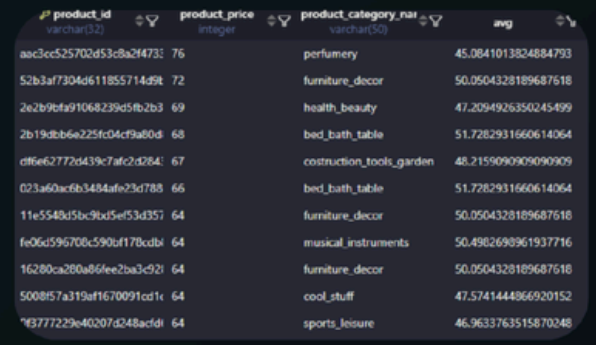
#11 - Based on the classification in the previous answer, summarize the number of products per each price category.

```sql
SELECT
CASE
WHEN product_price<=50 THEN 'Low Price'
WHEN product_price>50 AND product_price<100 THEN 'Medium Price'
ELSE 'High Price'
END AS price_category, count(distinct product_id) AS amount_products
FROM ecommerce_schema.products
GROUP BY 1
```

price_category	amount_products
High Price	610
Low Price	15414
Medium Price	16927

Amr Mousa

#12 -List of products (product id, product price, product category) and for each product, the average product price of the product category. Remove rows with NULL value in product price, and orderthe result based on product price descending.



product_id varchar(32)	product_price integer	product_category_nar varchar(50)	avg
aac3cc525702d53c8a2f4733	76	perfumery	45.084101382488479
52b3af7304d611855714d9b	72	furniture_decor	50.0504328189687618
2e2b9bfa91068239d5fb2b3	69	health_beauty	47.2094926350245499
2b19dbb6e225fc04cf9a80d	68	bed_bath_table	51.7282931660614064
dff6e62772d439c7afc2d284	67	costruction_tools_garden	48.2159090909090909
023a60acf6b3484afe23d788	66	bed_bath_table	51.7282931660614064
11e5548d5bc9bd5ef53d357	64	furniture_decor	50.0504328189687618
fe06d596708c590bf178cdb	64	musical_instruments	50.4982698961937716
16280ca280a86fee2ba3c92	64	furniture_decor	50.0504328189687618
5008f57a319af1670091cd1	64	cool_stuff	47.5741444866920152
f3777229e40207d248acfd	64	sports_leisure	46.9633763515870248

```sql
SELECT
product_id ,
product_price ,
product_category_name ,
avg(product_price) OVER (PARTITION BY product_category_name)
FROM ecommerce_schema.products
WHERE product_price is NOT NULL
ORDER BY product_price DESC ;
```

Amr Mousa

#13 - List the five heaviest products (product id, product category, product weight) from each product category

```sql
SELECT *
FROM (
SELECT product_id, product_category_name, product_weight_g, ROW_NUMBER()
OVER (PARTITION BY product_category_name ORDER BY product_weight_g DESC) as
row_num
FROM ecommerce_schema.products
WHERE product_price IS NOT NULL
) as t
WHERE row_num <= 5;
```

Amr Mousa

#14 -Create a view called “customers_shipped_vw” for all customers (customer id, customer name, customer city, order status) with orders that are in “shipped” status

```sql
CREATE VIEW ecommerce_schema.customers_shipped_vw AS
(
SELECT c.customer_id, c.customer_name, c.customer_city, o.order_status
FROM ecommerce_schema.customers as c
INNER JOIN ecommerce_schema.orders as o ON c.customer_id = o.customer_id
WHERE o.order_status = 'shipped'
);
```

Amr Mousa

#15 - Create a CTE (“with”) that will create a sequential number based on the product price for eachproduct category (product id, product category, product price, row num) and remove rows with NULL value in the product price.
Use the CTE to query for the three most expensive products from each category.

```sql
WITH product_category_CTE
AS
(
SELECT product_id, product_category_name, product_price, ROW_NUMBER() OVER
(PARTITION
BY product_category_name ORDER BY product_price DESC) as row_num
FROM ecommerce_schema.products
WHERE product_price IS NOT NULL
)
SELECT *
FROM product_category_CTE
WHERE row_num <= 3
```

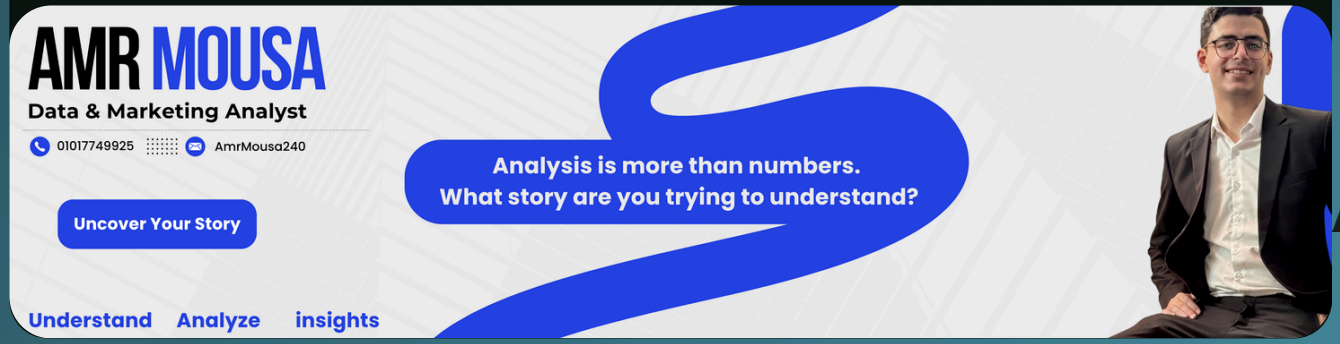
Amr Mousa

Level 2 – the end ..

>> The best SQL developers aren't the ones who know the most

>> They're the ones who practice the most

>> Skill grows quietly, One query at a time..



Practice

Reflect

Repeat