

SQL TESTS

IN DATA SCIENCE INTERVIEWS



WHAT TO EXPECT...

DATA SCIENCE INFINITY

You will often be provided with several key data tables (or sometimes just the table structures...)



products

product_id	product_name
181	Milk
427	Bread
211	Beer
...	...

customers

customer_id	country	age
AAA	UK	47
BBB	Germany	19
CCC	USA	29
...

transactions

customer_id	transaction_id	product_id	sales
AAA	001	427	34.22
AAA	001	665	9.99
CCC	003	211	101.87
...	...		



Tasks start easy...

...then get harder

Basic tasks...

Task	SQL Clauses to know...
Selection & Filtering	SELECT + WHERE
Finding Unique Values	DISTINCT + COUNT DISTINCT
Merging Multiple Tables	JOINS (LEFT, INNER, FULL)
Aggregating & Summarising	SUM, MAX, COUNT (+ GROUP BY)
Ordering	ORDER BY
Appending	UNION, UNION ALL

More complex tasks...

Task	SQL Clauses to know...
Working with dates	DATEDIFF, DATEADD, DATETRUNC
Conditional Logic	CASE WHEN
Apply logic to a set of rows	RANK, NTILE, LAG, LEAD (Window Functions)
Combinations	CROSS JOIN



Don't forget about **TEMP**
TABLES or **CTE**! These
might make solving the
task possible (or just
make your life easier!)



Read each question
carefully - think
about *exactly* what it
is asking, because...



Near the end, they may
try to lure you into an
obvious solution...where in
reality, more thought is
required!

For example...

Q) "Find the highest value for column X after joining Table A & Table B"

This may seem easy, but your query could return multiple rows if there was a tie in the data - is this ok?!



If you don't know the exact
syntax - don't leave the
question blank...

Write down what you think
should happen in words!

Example Test - Data

For this test, we have 5 tables that can be accessed in the *grocery_db* schema of the DATA SCIENCE INFINITY database. Example data from each table can be seen below...

product_areas

product_area_id	product_area_name	profit_margin
1	Non-Food	0.25
2	Vegetables	0.18
3	Fruit	0.14
4	Dairy	0.19
5	Meat	0.11

loyalty_scores

customer_id	customer_loyalty_score
104	0.587
69	0.156
525	0.959
181	0.418
796	0.57

customer_details

customer_id	distance_from_store	gender	credit_score
754	1.17	M	0.75
843	4.84		
749	1.74	M	0.65
426	4.38	F	0.57
560		M	0.54

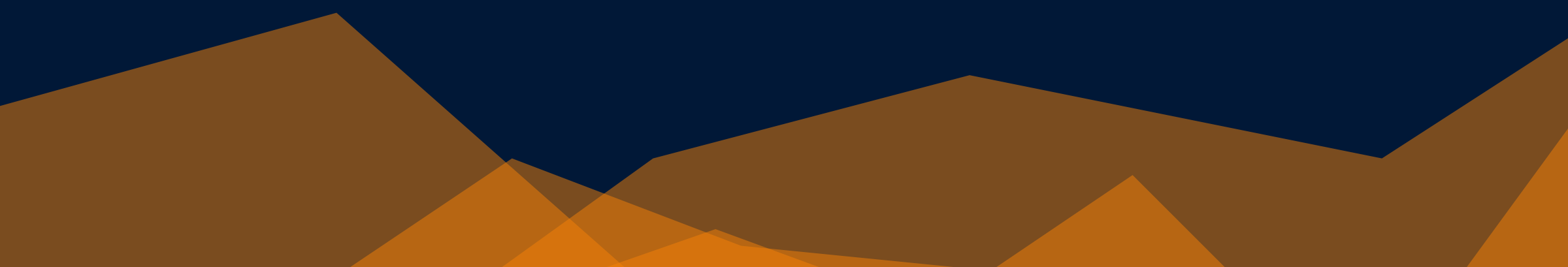
transactions

customer_id	transaction_date	transaction_id	product_area_id	num_items	sales_cost
642	2020-04-01	435561233435	4	3	9.44
642	2020-04-01	435561233435	3	5	23.82
493	2020-07-15	436618008621	4	1	6.83
493	2020-07-15	436618008621	3	9	9.33
493	2020-07-15	436618008621	5	1	8.50

campaign_data

customer_id	campaign_name	campaign_date	mailer_type	signup_flag
74	delivery_club	2020-07-01	Mailer2	1
655	delivery_club	2020-07-01	Mailer2	0
607	delivery_club	2020-07-01	Mailer2	1
788	delivery_club	2020-07-01	Control	0
405	delivery_club	2020-07-01	Mailer1	0

Example Test - Questions

- 1) How many rows are there in the *transactions* table?
 - 2) Return the *customer_id* for the customer who lives farthest from the store
 - 3) Return the number of unique customers in the *customer_details* table, split by *gender*
 - 4) Return the total sales for each *product_area_name* for July 2020 – in the order of highest sales to lowest sales
 - 5) For the customers with a *customer_loyalty_score*, divide them up into 10 deciles, and calculate the average *distance_from_store* for each decile
 - 6) Return a list of all *customer_id*'s that DO NOT have a *loyalty_score* (i.e. they are in the *customer_details* table, but not in the *loyalty_scores* table)
 - 7) Return data showing, for each *product_area_name* – the total sales, and the percentage of overall sales that each *product_area* makes up
- 
- The bottom of the slide features a decorative graphic consisting of several overlapping, semi-transparent geometric shapes in shades of brown and orange, creating a layered, mountain-like effect.