#Digital\_Data\_Query\_Assign

SELECT \*

FROM USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1;

select

count(distinct cookie\_id)Unique\_Users,

count(page\_session\_id)Pageviews,

count(page\_session\_id)/count(distinct cookie\_id)Avg\_Pageviews\_per\_User

from USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1

select

count(distinct cookie\_id)Unique\_Users,

sum(engaged\_time\_on\_page\_seconds)engview,

sum(engaged\_time\_on\_page\_seconds)/count(distinct cookie\_id)Avg\_eng\_per\_User

from USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1

SELECT TO\_CHAR(DATEADD(second, LAST\_PING\_TIMESTAMP, '1970-01-01'::timestamp), 'DD-MM-YYYY') AS date\_in\_IST

FROM USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1;

SELECT COOKIE\_ID,

CONVERT\_TIMEZONE('UTC', 'Asia/Kolkata', TO\_TIMESTAMP(LAST\_PING\_TIMESTAMP)) AS timestamp\_in\_ist

FROM USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1;

SELECT distinct COOKIE\_ID,

TIME\_FROM\_PARTS(

EXTRACT(HOUR FROM CONVERT\_TIMEZONE('UTC', 'Asia/Kolkata', TO\_TIMESTAMP(LAST\_PING\_TIMESTAMP))),

EXTRACT(MINUTE FROM CONVERT\_TIMEZONE('UTC', 'Asia/Kolkata', TO\_TIMESTAMP(LAST\_PING\_TIMESTAMP))),

EXTRACT(SECOND FROM CONVERT\_TIMEZONE('UTC', 'Asia/Kolkata', TO\_TIMESTAMP(LAST\_PING\_TIMESTAMP)))

) AS time\_in\_ist

FROM USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1;

-- select DATE(LAST\_PING\_TIMESTAMP) as LAST\_PING\_DATE from USER\_DATA\_DIGITAL.DIGITAL\_DATA.DEMO\_DATA\_1;

select TO\_CHAR(TO\_TIMESTAMP(LAST\_PING\_TIMESTAMP), 'DD-MM-YYYY') as PING\_DATE FROM USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1;

-- SELECT COOKIE\_ID,

-- HOUR(CONVERT\_TIMEZONE('UTC', 'Asia/Kolkata', TO\_TIMESTAMP(LAST\_PING\_TIMESTAMP))) AS hour\_in\_ist

-- FROM USER\_DATA\_DIGITAL.DIGITAL\_DATA.DEMO\_DATA\_1;

SELECT

TO\_CHAR(TO\_TIMESTAMP(LAST\_PING\_TIMESTAMP), 'DD-MM-YYYY') AS LAST\_PING\_DATE,

EXTRACT(HOUR FROM TO\_TIMESTAMP(LAST\_PING\_TIMESTAMP)) AS ENGAGEMENT\_HOUR

FROM

USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1;

ALTER TABLE USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1

ADD COLUMN PING\_DATE\_CHANGED VARCHAR(10);

UPDATE USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1

SET PING\_DATE\_CHANGED = TO\_CHAR(TO\_TIMESTAMP(LAST\_PING\_TIMESTAMP), 'DD-MM-YYYY');

SELECT cookie\_id, DATE\_PART(HOUR,TO\_TIMESTAMP(last\_ping\_timestamp)) AS HOUR

from USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1;

SELECT EXTRACT (HOUR FROM TO\_TIMESTAMP(LAST\_PING\_TIMESTAMP))

FROM USER\_DATA\_DIGITAL.DIGITAL\_DATA.DEMO\_DATA\_1;

select TO\_CHAR(TO\_TIMESTAMP(LAST\_PING\_TIMESTAMP), 'HH') as Average\_Hours\_Spent from USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1;

SELECT (COUNT(DISTINCT CASE WHEN DEVICE = 'desktop' THEN COOKIE\_ID END) \* 100.0 / COUNT(DISTINCT COOKIE\_ID)) AS desktop\_usage\_percentage

FROM USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1;

SELECT COUNT(DISTINCT cookie\_id)android\_users

FROM USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1

WHERE user\_agent LIKE '%Android%';

SELECT COUNT(DISTINCT cookie\_id)registered\_count

FROM USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1

WHERE SUBSCRIBER\_ACCT = 1;

WITH section\_counts AS (

SELECT SECTION2, COUNT(DISTINCT cookie\_id) AS user\_count

FROM USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1

GROUP BY SECTION2

)

SELECT SECTION2, user\_count

FROM section\_counts

WHERE user\_count = (SELECT MAX(user\_count) FROM section\_counts);

SELECT SECTION2, COUNT(\*) AS frequency

FROM USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1

GROUP BY SECTION2

ORDER BY frequency DESC;

WITH user\_returning\_counts AS (

SELECT cookie\_id, MAX(CASE WHEN NEW\_USER = false THEN 1 ELSE 0 END) AS is\_returning

FROM USER\_DATA\_DIGITAL.DIGITAL\_DATA.DIGITAL\_DATA\_1

GROUP BY cookie\_id

)

SELECT AVG(is\_returning) AS avg\_returning\_users

FROM user\_returning\_counts;

-----

#Bronze to silver

CREATE OR REPLACE TABLE PWC\_SILVER.DIGITAL\_SILVER.DIGITAL\_SILVER\_T1

AS SELECT

LAST\_PING\_TIMESTAMP,

COOKIE\_ID,

PAGE\_SESSION\_ID,

PATH,

TO\_CHAR(CONVERT\_TIMEZONE('UTC', 'Asia/Kolkata', TO\_TIMESTAMP(LAST\_PING\_TIMESTAMP)), 'DD-MM-YYYY HH24:MI:SS "IST"') AS Formatted\_Date,

TO\_CHAR(TO\_TIMESTAMP(LAST\_PING\_TIMESTAMP), 'DD-MM-YYYY') as SUB\_DATE

FROM PWC\_BRONZE.DIGITAL\_BRONZE.DIGITAL\_BRONZE\_T1;

SELECT \*

from PWC\_SILVER.DIGITAL\_SILVER.DIGITAL\_SILVER\_T1;

-----

Silver\_Day6

create or replace table SALES\_SILVER as

select \* from PWC\_BRONZE.BRONZE.SALES\_BRONZE;

create or replace table CUSTOMER\_SILVER as

select \* from PWC\_BRONZE.BRONZE.CUSTOMER\_BRONZE;

alter table SALES\_SILVER

add column TOTAL\_PRICE NUMBER;

update SALES\_SILVER

set TOTAl\_PRICE=QUANTITY\*PRICE;

DESC table SALES\_SILVER;

alter table SALES\_SILVER

add column ORDERYEAR NUMBER;

update SALES\_SILVER

set ORDERYEAR=YEAR(ORDERDATE);

select \* from CUSTOMER\_SILVER;

update CUSTOMER\_SILVER set CUSTOMERNAME=UPPER(CUSTOMERNAME);

select c.customername, c.customerid,s.orderid,c.country,s.quantity, s.total\_price

from PWC\_SILVER.DAY6\_PWC.SALES\_SILVER s

join PWC\_SILVER.DAY6\_PWC.CUSTOMER\_SILVER c

on s.customerid = c.customerid;

select \* from gold\_summary;

-- Write a udf which gives all the details of the sales under the a particular country

----

# Day7\_UDF

create or replace function get\_Date(business\_date timestamp)

returns date

language sql

as

$$

TO\_DATE(SUBSTR(TO\_CHAR(business\_date),1,10))

$$;

select get\_Date('2024-06-18 17:53:22.000')

CREATE OR REPLACE FUNCTION join\_table()

RETURNS TABLE (

customerid NUMBER,

customername VARCHAR,

orderid VARCHAR,

orderyear NUMBER

)

language sql

as

$$

SELECT c.customerid, c.customername, s.\*,

FROM PWC\_SILVER.DAY6\_PWC.SALES\_SILVER s

JOIN PWC\_SILVER.DAY6\_PWC.CUSTOMER\_SILVER c

ON s.customerid = c.customerid

WHERE c.country = country\_name

$$;

SELECT \* FROM TABLE(get\_sales\_by\_country('USA'));

----

#Day8\_Procedure

create or replace PROCEDURE eliminate\_orderid()

returns VARCHAR

language SQL

AS

$$

DECLARE

message VARCHAR;

BEGIN

DELETE FROM SALES\_SILVER WHERE ORDERID = '3';

message := 'Data Deleted';

RETURN message;

END;

$$

;

CALL eliminate\_orderid();

CREATE OR REPLACE PROCEDURE eliminate\_orderid(orderid\_to\_delete VARCHAR)

RETURNS VARCHAR

LANGUAGE SQL

AS

$$

DECLARE

message VARCHAR;

BEGIN

DELETE FROM SALES\_SILVER WHERE ORDERID =: orderid\_to\_delete;

message := 'Order ID ' || orderid\_to\_delete || ' deleted successfully.';

RETURN message;

END;

$$;

CALL eliminate\_orderid(2);

CREATE OR REPLACE PROCEDURE eliminate\_orderid(orderid\_to\_delete VARCHAR)

RETURNS VARCHAR

LANGUAGE SQL

AS

$$

DECLARE

message VARCHAR;

BEGIN

DELETE FROM SALES\_SILVER WHERE ORDERID =: orderid\_to\_delete;

RETURN 'Order ID ' || orderid\_to\_delete || ' deleted successfully.';

END;

$$;

CALL eliminate\_orderid(1);

--weekend assignment

-- create a doc over the weekend with all things mentioned

-- execute procedure and task in all bronze, silver & gold

CREATE OR REPLACE PROCEDURE push\_bronze\_to\_silver()

RETURNS VARCHAR

LANGUAGE SQL

AS

$$

DECLARE

message VARCHAR;

BEGIN

INSERT INTO PWC\_SILVER.DAY8\_PWC.NEW\_SALES\_SILVER (

SELECT \*

FROM PWC\_BRONZE.BRONZE.SALES\_BRONZE

);

RETURN 'Data pushed successfully';

END;

$$;

CALL push\_bronze\_to\_silver();

-- procedure for doing both (bronze to silver & silver ot gold in one code only)

CREATE OR REPLACE PROCEDURE push\_bronze\_to\_silver\_then\_gold()

RETURNS VARCHAR

LANGUAGE SQL

AS

$$

DECLARE

message VARCHAR;

BEGIN

BEGIN

INSERT INTO PWC\_SILVER.DAY8\_PWC.NEW\_SALES\_SILVER (

SELECT \*

FROM PWC\_BRONZE.BRONZE.SALES\_BRONZE

);

RETURN 'Data pushed successfully. \n';

END;

BEGIN

INSERT INTO PWC\_GOLD.DIGITAL\_GOLD.FOLD\_SUMMARY (

SELECT \*

FROM PWC\_SILVER.DAY8\_PWC.NEW\_SALES\_SILVER

);

RETURN 'Data appended successfully from NEW\_SALES\_SILVER (silver) to SALES\_GOLD (gold).\n';

END;

END;

$$;

CALL push\_bronze\_to\_silver\_then\_gold();

-- task for pushing data from bronze to silver every night at 12 PST

CREATE OR REPLACE TASK push\_bronze\_to\_silver\_task

WAREHOUSE = 'COMPUTE\_WH'

SCHEDULE = 'USING CRON 30 20 \* \* \* UTC' --runs at midnight

AS

CALL push\_bronze\_to\_silver();

--task for pushing data simultaneously from bronze to silver and then gold

CREATE OR REPLACE TASK push\_bronze\_to\_silver\_to\_gold

WAREHOUSE = 'COMPUTE\_WH'

SCHEDULE = 'USING CRON 30 20 \* \* \* UTC' --runs at midnight

AS

CALL push\_bronze\_to\_silver\_then\_gold();

----

# Day9\_Timestamp

SHOW PARAMETERS LIKE 'data\_retention\_time\_in\_days' IN DATABASE PWC\_BRONZE;

ALTER DATABASE PWC\_BRONZE SET DATA\_RETENTION\_TIME\_IN\_DAYS = 50;

select \* from customer\_bronze;

SHOW PARAMETERS LIKE 'data\_retention\_time\_in\_days' IN TABLE PWC\_BRONZE.BRONZE.CUSTOMER\_BRONZE;

DROP TABLE customer\_bronze;

UNDROP TABLE customer\_bronze;

DELETE FROM customer\_bronze

where customerid = '103';

select \*

from customer\_bronze

BEFORE (OFFSET => - 10\*60);

SELECT \* from customer\_bronze BEFORE (STATEMENT => '01b53ee1-0001-2587-0004-b92600013dfe');

--assignment

--use the timestamp clause

--use cloning feature in place of creating new table