**SQL CHEATCODE --refer datalemur and practice nice amount of questions  
check for extract and to char fn**

MySQL powers many of the most accessed applications, including Facebook, Twitter, Netflix, Uber, Airbnb, Shopify, and Booking.com.

Ddl commands: create, alter, drop, truncate  
  
CREATE TABLE SCHOOL(  
 SCHOOL\_CODE INT NOT NULL,  
 SCHOOL\_NAME VARCHAR(500),  
 ADDRESS VARCAHR(500),  
 PHONE\_NUMBER INT,  
 STARS INT,  
 AGE INT,  
 PRIMARY KEY (SCHOOL\_CODE)  
);

ALTER TABLE SCHOOL ADD CITY VARCHAR(100); -- adding column in existing table

ALTER TABLE SCHOOL DROP AGE; -- drop the column from existing table

ALTER TABLE SCHOOL RENAME STARS TO RATING; --renaming column name

ALTER TABLE SCHOOL MODIFY COLUMN SCHOOL\_CODE VARCHAR(10);

DROP TABLE SCHOOL;

TRUNCATE TABLE SCHOOL;

CREATE TABLE TEACHERS(  
 STAFF\_ID INT NOT NULL PRIMARY KEY,  
 STAFF\_NAME VARCHAR(50),  
 SPEACIAL\_ACTIVITY VARCHAR(50),  
 SCHOOL\_CODE VARCHAR(10) FOREIGN KEY REFERENCES SCHOOL(SCHOOL\_CODE)  
);  
  
CREATE TABLE CULTURAL\_ACTIVITY(  
 SERIAL\_NO INT AUTO\_INCREMENT,  
 ACTIVITY UNIQUE NOT NULL,  
 CATEGORY VARCHAR(20) CHECK(CATEGORY IN (‘ENTERTAINMENT’,’SPORTS’)  
);

CREATE TABLE NEW\_STUDENT(  
 STUDENT\_ID VARCAHAR(20) PRIMARY KEY,  
 NAME VARCHAR(200),  
 ACTIVITY\_PARTICIPATED VARCHAR(200),  
 ACHIEVEMENT\_TAB VARCHAR(20),  
 SCHOOL\_CODE VARCHAR(10) FOREIGN KEY REFERENCES SCHOOL(SCHOOL\_CODE),  
 COACH\_ID VARCHAR(10) FOREIDN KEY REFERENCES TEACHERS(STAFF\_ID)  
);

DML commands : Insert,update,delete, merge  
INSERT INTO SCHOOL (SCHOOL\_CODE,SCHOOL\_NAME,ADDRESS,PHONE\_  
NUMBER,RATING) VALUES (‘HS394’,’ TMHNU.Mat.Hr.Sec.School’,’ KMC Nagar,Theni’, 9943565608,3);  
INSERT INTO SCHOOL (SCHOOL\_CODE,SCHOOL\_NAME,ADDRESS,PHONE\_  
NUMBER,RATING) VALUES(‘SS762’,’ VSJR.Sec.School’,’ Lakshmi Housing Street, Karur’, 8344524956, 5);  
INSERT INTO SCHOOL (SCHOOL\_CODE,SCHOOL\_NAME,ADDRESS,PHONE\_  
NUMBER,RATING) VALUES(‘HS994’,’Meenakshi.Govt.Hr.Sec.School’,’ Samudhaya Aram Street, Vellore’ , 9876768239,4);  
INSERT INTO SCHOOL (SCHOOL\_CODE,SCHOOL\_NAME,ADDRESS,PHONE\_  
NUMBER,RATING) VALUES(‘SS834’,’ KCT.Sec.School’,’ Saravanampatti,Cbe’,7890654312,2);  
INSERT INTO SCHOOL (SCHOOL\_CODE,SCHOOL\_NAME,ADDRESS,PHONE\_  
NUMBER,RATING) VALUES(‘SS882’,’ Kgsil.Public.Sec.School’,’ Keeranatham, Cbe’, 6584768323,’’);  
INSERT INTO SCHOOL (SCHOOL\_CODE,SCHOOL\_NAME,ADDRESS,PHONE\_  
NUMBER,RATING) VALUES(‘HS950’,’ Ruby Mat.Hr.Sec.School’,’ Vikas Avenue, Madurai’, 6380464655,5)  
---  
  
DELETE FROM GFG\_Employees WHERE NAME = 'Rithvik'

CREATE TABLE NEW\_TABLE AS SELECT \* FROM SCHOOL  
OR   
CREATE TABLE NEW\_SCHOOL\_DATA AS SELECT SCHOOL\_CODE,SCHOOL\_NAME FROM SCHOOL  
WHERE COUNTRY =’  
OR   
SELECT \* INTO NEW\_TABLE FROM OLD\_TABLE

UPDATE SCHOOL  
SET SCHOOL\_NAME =’w3schools’  
where school\_code =’HS200’;

Coalesce and ifnull

For eg arg1= null, arg2 = 15, arg3 =13   
both does similar function but in coalesce it gives 1st not null value  
in if null it deals with only two values

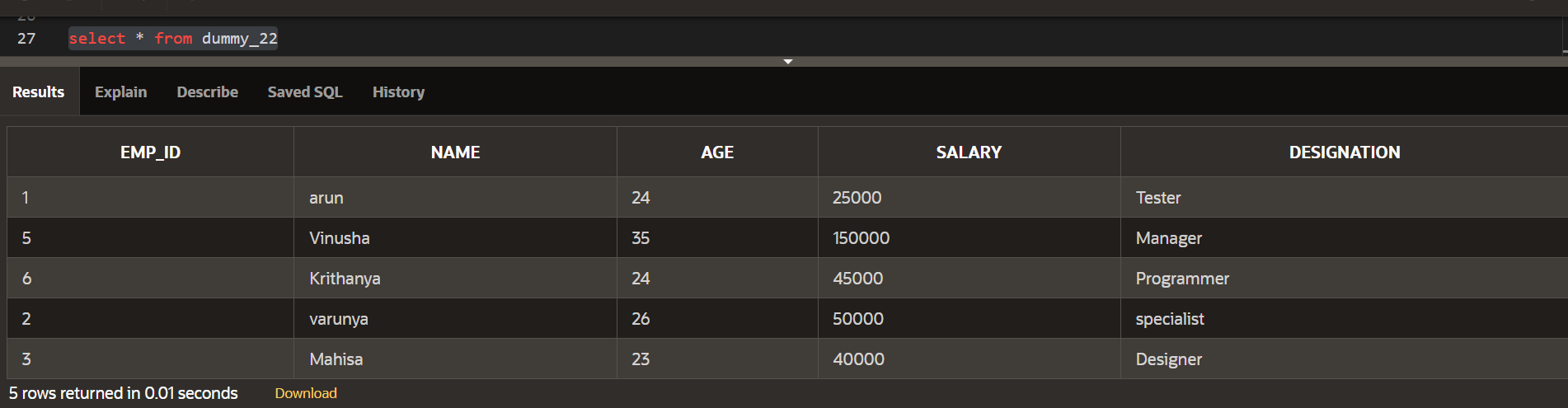
Merge performs operation like insert, update and delete together  
 It need to have source table and target table. In source table updated table or list will be there whereas in target table where the changes need to be implemented

merge syntax:

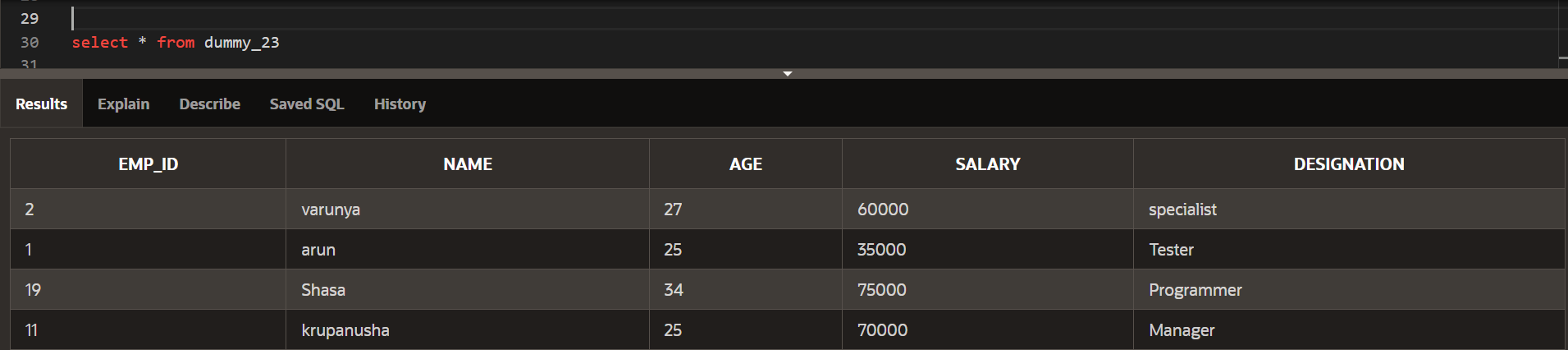
A screenshot of a computer

Description automatically generated

Merge demo:

Target table  


Source table



merge into dummy\_22 t   
using dummy\_23 s  
on (t.emp\_id = s.emp\_id)  
when matched then   
 update set t.age = s.age, t.salary = s.salary  
when not matched then  
 insert(t.emp\_id,t.name,t.age,t.salary,t.designation) values(s.emp\_id,s.name,s.age,s.salary,s.designation)  
when not matched by source then   
 delete;

-------------------count filter clause-----------------

SELECT

COUNT(\*) FILTER (WHERE conditional\_expression)

FROM table\_name;

example

SELECT

COUNT(\*) FILTER (WHERE device\_type = 'laptop') AS laptop\_views,

COUNT(\*) FILTER (WHERE device\_type IN ('tablet', 'phone')) AS mobile\_views

FROM viewership;

Sql general codes:  
   
select review\_id, submit\_id, stars  
from reviews  
where stars < 4 and review\_id%2=0  
--------------  
SELECT \* FROM reviews

WHERE NOT rating = 5;  
---------------

SELECT \* FROM reviews

WHERE stars NOT BETWEEN 2 AND 4;  
---------------------  
SELECT drug, manufacturer, units\_sold

FROM pharmacy\_sales

WHERE manufacturer IN ('Biogen', 'Bayer', 'Eli Lilly');  
-------------------------  
select product\_id, drug from pharmacy\_sales  
where drug like ‘%relief%’  
Arithmetic functions: it deals with symbols like +,-,\*,/

Where math functions are floor,ceil,round,abs,trunc  
floor- round down, ceil – round up, abs – it will give difference value without - symbol  
trunc – it will not roundup the value  
:: - this notation denotes to type cast

SELECT

character,

superhero\_alias,

platform,

CASE

WHEN followers >= 500000 THEN 'Popular'

END AS popularity\_category

FROM marvel\_avengers;

-----Joins------  
SELECT

orders.order\_id,

deliveries.delivery\_id,

deliveries.delivery\_date,

deliveries.delivery\_status

FROM orders

LEFT JOIN deliveries

ON orders.order\_id = deliveries.order\_id;

----Date function-----------  
SELECT  
 message\_id,   
 sent\_date,  
 EXTRACT(YEAR FROM sent\_date) AS extracted\_year,  
 DATE\_PART('year', sent\_date) AS part\_year,  
from messages

the art of dealing people