

08/05/24

SQL 03': CRUD - 1

AGENDA

- ① Sakila DB
- ② INSERT
- ③ SELECT
- ④ DISTINCT
- ⑤ WHERE
- ⑥ AND, OR, NOT, IN
- ⑦ Operations on COL, ALIAS
- ⑧ BETWEEN.

C	-	CREATE	(INSERT)
R	-	READ	(SELECT)
U	-	UPDATE	(UPDATE)
D	-	DELETE	(DELETE)

CREATE / INSERT

```
INSERT INTO TABLE-NAME (C1, C2, C3...)
VALUES (V1, V2, V3...);
```

);

Dem

```
(C1, C2, C3)
(V1, V2, V3)
(V2, V3, V1)
```

```
INSERT INTO TABLE  
VALUES (V1, V2, V3, ...),  
      - - -  
      - - -  
);
```

V2, V1, V3, V5, ...)

```
CREATE TABLE TABLE-NAME (  
    C1 - -  
    C2 . - -  
    C3 . -  
    C4 . -  
);
```

NOTE: Always write the column names in INSERT

- ① Less prone to errors
- ② Less tedious to write
- ③ Definitely give all the values.

We can skip values when: -

- ① DEFAULT
- ② NULL
- ③ AUTO-INCR

Pseudo code (DISTINCT)

```
Select TITLE, DISTINCT REL-YR  
From FILM;
```

ANIMAL	2024	2024
POSHPA	2024	
:		
:		

ans = []

for each row in film :
 ans.add(row);

filt-ans = []

for each row in ans :
 filt-ans.add (row ['rel-yr'],
 row ['rating'])

final-ans = set (filt-ans) ^{usr} [1, 1, 3, 2, 2, 5, 6]

print (final-ans) [1, 2, 3, 5, 6]

(2024 , PG-13)
(2024 , PG-13)
(2024 , PG)
(2024 , R)

BREAK TILL - 8:25 AM.

Q. Print the name of films released in year=2024.

WHERE

Pseudo Code (WHERE)

```
1 ans = []
2
3 for row in film:
4     if (row matches WHERE cond'n):
5         ans.add(row)
6
7
8 filt-ans = []
9 for row in ans:
10    filt-ans.add(row['title'])
11
12
13
14 print(filt-ans)
```

if year = 2024

Note: filter early.

DISTINCT + WHERE

```
ans = []
for each row in table-name:
    if (row matches WHERE cond'n):
        ans.add(row)
```

From

WHERE

```
filt-ans = []  
for each row in ans: ← SELECT  
    filt-ans.add (row['year'],  
                  row['rating'])
```

```
final-ans = set (filt-ans) ← DISTINCT.  
print (final-ans)
```

WHERE

Q. print all film names ~~2024~~ AND PG-13

Revision

- ① INSERT
- ② SELECT
- ③ DISTINCT
- ④ WHERE
- ⑤ ALIAS
- ⑥ AND, OR, NOT, IN
- ⑦ BETWEEN.

Next class

(CONCAT, IFNULL,
SUBSTR)

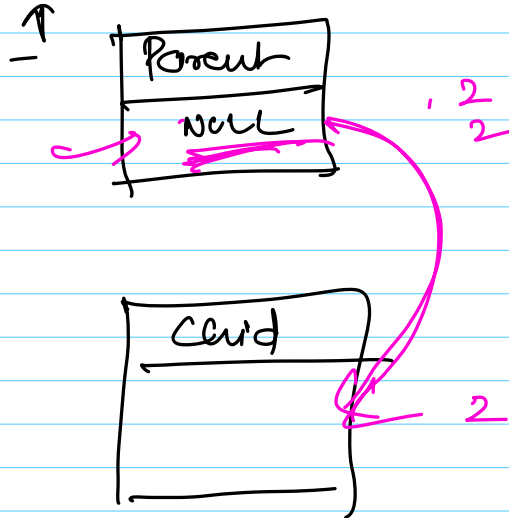
- ① LIKE
- ② IS NULL
- ③ ORDER BY
- ④ UPDATE
- ⑤ DELETE
↳ DELETE, TRUNCATE, DROP.

UNIQUE

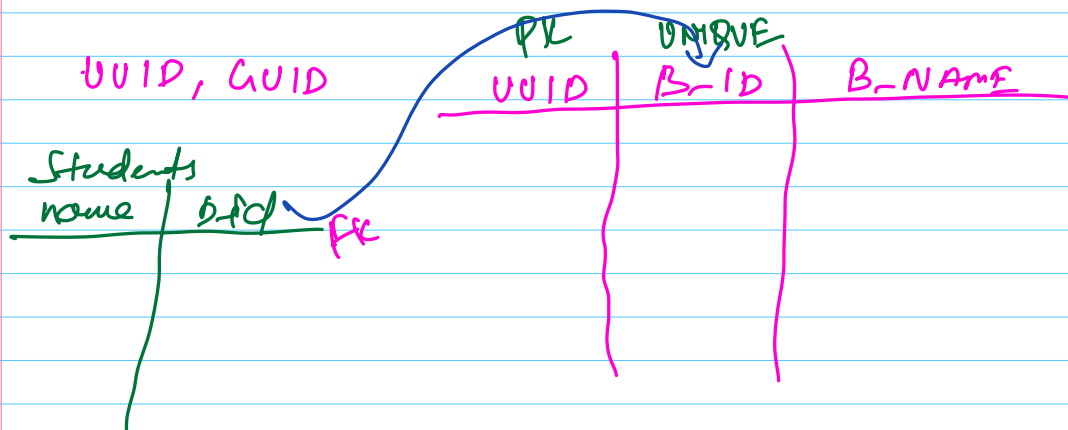
NOT NULL

ON DELETE CASCADE

DUPLICATES



PK = (UNIQUE + NOT NULL)



QUERIES RUN IN CLASS

-- SQL 03 CRUD 1

USE SAKILA;

INSERT INTO film (title, description, release_year, language_id, rental_duration, rental_rate, length, replacement_cost, rating, special_features)

VALUES ('KANTARA', 'Batman fights the Joker', 2022, 1, DEFAULT, DEFAULT, 152, 19.99, 'PG-13', 'Trailers'),

('DEADPOOL', 'Batman fights Bane', 2024, 1, 5, 5.99, 165, 19.99, 'PG-13', 'Trailers'),

('GADAR', 'Batman fights Superman', 2023, 1, 7, 7.99, 152, 19.99, 'PG-13', 'Trailers');

INSERT INTO film (title, description, release_year, language_id, length, replacement_cost, rating, special_features)

VALUES ('WOLVERINE', 'Batman fights the Joker', 2022, 1, 152, 19.99, 'PG-13', 'Trailers');

SELECT * FROM FILM;

SELECT 1 FROM FILM;

SELECT "HELLO RAUSHAN" FROM FILM;

SELECT 1+1 ;

SELECT 2*5 AS 2X5;

SELECT 1+1 FROM FILM;

SELECT 1+1 FROM DUAL;

-- OPERATIONS ON COLUMN

SELECT TITLE, LENGTH FROM FILM;

SELECT TITLE, LENGTH/60 AS HRS FROM FILM;

SELECT TITLE, ROUND(LENGTH/60) AS HRS FROM FILM;

SELECT TITLE, FLOOR(RENTAL_DURATION/(LENGTH/60)) AS TIMES_CAN_WATCH FROM FILM;

-- LIST ALL UNIQUE YEARS THAT HAVE A FILM

SELECT DISTINCT RELEASE_YEAR FROM FILM;

SELECT DISTINCT RATING FROM FILM;

-- (2024 , PG-13) , (2024, R) ...

```
SELECT DISTINCT RELEASE_YEAR, RATING FROM FILM;
```

```
-- WHERE  
SELECT TITLE FROM FILM  
WHERE RELEASE_YEAR=2024;
```

```
-- AND  
SELECT TITLE FROM FILM  
WHERE RELEASE_YEAR=2024 AND RATING = 'PG-13';
```

```
-- OR  
SELECT TITLE FROM FILM  
WHERE RELEASE_YEAR=2024 OR RATING = 'PG-13';
```

```
--
```

```
--
```

```
SELECT TITLE, RELEASE_YEAR FROM FILM  
WHERE RELEASE_YEAR=2024 OR RELEASE_YEAR=2022 OR RELEASE_YEAR=2016;
```

```
SELECT TITLE, RELEASE_YEAR FROM FILM  
WHERE RELEASE_YEAR IN (2024, 2022, 2016);
```

```
SELECT TITLE, RELEASE_YEAR FROM FILM  
WHERE RELEASE_YEAR NOT IN (2024, 2022, 2016);
```

```
-- NOT  
SELECT TITLE, RELEASE_YEAR FROM FILM  
WHERE RELEASE_YEAR != 2006;
```

```
SELECT TITLE, RELEASE_YEAR FROM FILM  
WHERE RELEASE_YEAR <> 2006;
```

```
SELECT TITLE, RELEASE_YEAR FROM FILM  
WHERE NOT RELEASE_YEAR=2006 ;
```

```
-- RELEASE BETWEEN 2022 AND 2024 (BOTH INCLUSIVE)  
-- BETWEEN
```

```
SELECT TITLE, RELEASE_YEAR FROM FILM  
WHERE RELEASE_YEAR >=2022 AND RELEASE_YEAR<=2024;
```

```
SELECT TITLE, RELEASE_YEAR FROM FILM  
WHERE RELEASE_YEAR BETWEEN 2022 AND 2024;
```

```
-- SELECT TO COPY DATA FROM TABLE  
CREATE TABLE FILM_COPY  
AS SELECT * FROM FILM;
```