

CRUD - II

Good morning everyone!

We start at 7:05 AM



Agenda .

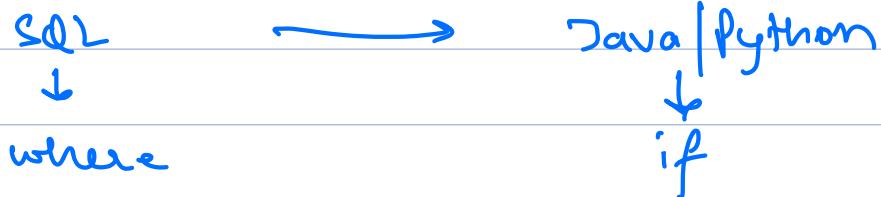
Read → AND/OR/NOT

- IN
- BETWEEN
- LIKE
- IS NULL
- ORDER BY
- LIMIT

}

Update]
Delete

AND operator .



Select * from table where id = 2 ;

if(id == 2)
ans.add(row)

if marks > 40 || marks < 60)

if (bsp > 80 || bsp < 90)

AND

Q Get all movies that have a rating of PG-13
and were released in 2006.

Select * from films where rating = 'PG-13';

Select * from films where release_year
= 2006;

→ if (rating == 'PG-13' && release_year == 2006)

Select * from film

where rating = 'PG-13'

AND release_year = 2006 ;

Q Get all movies that have a rating of PG-13 OR were released in 2006.

→ if (rating == 'PG-13' || release-year == 2006)

Select * from films
where rating = 'PG-13'
OR release-year = 2006 ;

Q Get the movies with rating anything other than PG-13 **and** release year any other than 2006;

Rating	Year
PG-13	2005 X
R	2006
PG-13	2006
A	2022

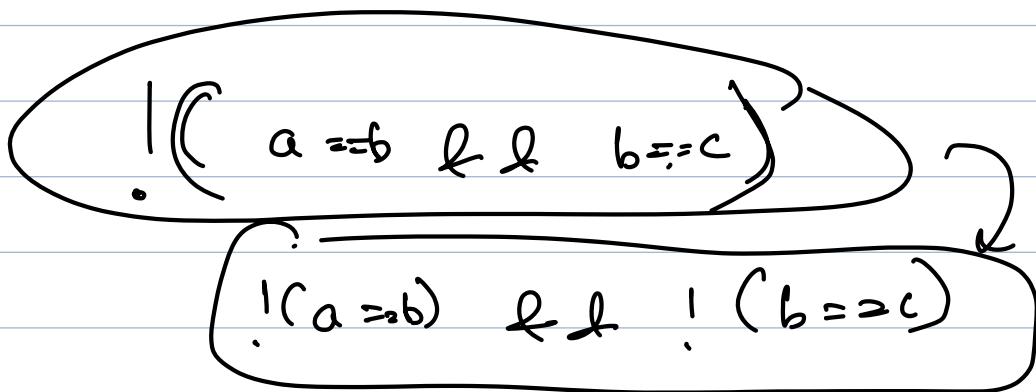
Select * from film
where (NOT rating = 'PG-13')
AND (NOT release-year = 2006)

select * from film
where rating = 'PG-13'
AND NOT release-year = 2006

NOT (rating = 'PG-13')

NOT

release-year = 2006)



* Always use parenthesis whenever combining multiple conditions.

!= \Rightarrow not equal in coding

!= or <> \Rightarrow not equal in SQL

select * from film
where rating != 'PG-13'
AND ARID release-year != 2006;
OR

select * from film
where rating <> 'PG-13'

and release_year <> 2006;

\neq \neq
 \ / / \
NOT EQUAL

Coding vs SQL

$==$	$=$
$<$	$<$
\leq	\leq
$>$	$>$
\geq	\geq
\neq	\neq or \neq

IN operator.

Students		
b_id	name	b_id

Q get all the students of b_id 1, 3, 5 or 7.

select * from students

where b_id = 1

or b_id = 3

or b_id = 5

or b_id = 7 ;

$\text{IN} \rightarrow$ when you want to compare value of one column against multiple values.

select * from students
where b_id IN (1, 3, 5, 7);

BETWEEN Operator.

Q) movies released b/w 2005 - 2010
students with bsp from 50 - 70

select * from students
where bsp >= 50
and bsp <= 70;

students \rightarrow bsp $(50 - 70)$
 $\& (80 - 95)$

select * from students
where (bsp >= 50 AND bsp <= 70)
~~AND (bsp >= 80 AND bsp <= 95) ;~~
OR

$>= x$ AND $<= y$
 \Downarrow

BETWEEN

Select * from students
where (psb between 50 AND 70)
OR (psb between 80 AND 95);

LIKE operator

title (string)

(Ujjwal is batman

batman begin

Hello brother

title = 'batman begin')

batches.

id	name
Academy	Apo 22 Beg Java
May Academy	Python

- Every Academy batch should have 'academy' in its name
- Every beg batch should have 'beg' in its name.

- every morning batch should have 'morning' in its name.

Apr 22 beg morning academy

academy morning Apr 22 beg

Q return all morning batches in academy.

```
boolean isMorningAcademy ( String batchName ) {  
    if( batchName .contains ( morning ) &&  
        batchName .contains ( academy ))  
        return true ;  
    return false ;  
}
```

Like → when you have to check inside
a string

2 wild card symbols:

① '-' → exactly one occurrence of any character.

② '.' → any no. of occurrences of any character

'.' → 'ab' - '123'
→ 'abc' - 'alpha'
→ 'aab' -
→ ' ' -
→ 'xyz' -

- → a
b
l
b

-- → ab
xy
al
bz
(a) X

12
15

abc ✓
xyz ✓
aaa ✓
ab X
a X
abl ✓
ab12 X

$\%$ → any char any no. of time

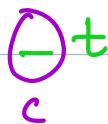
$-$ → any char  time

String

Cat

patterns

$_ \cdot t$
r r r
c a

 t
c

a
 +

$\% \cdot t$
r
ab
vij
ca

$\% \text{ cat } \%$
T T
- -

cat

title

Q To check if ~~hello~~ exists in selected column

where title like '%. hello %' ;

Hello Brother

Brother hello mine

'/.hello %

Bro he wello

he lb %.

bw hello

01 hello 1.

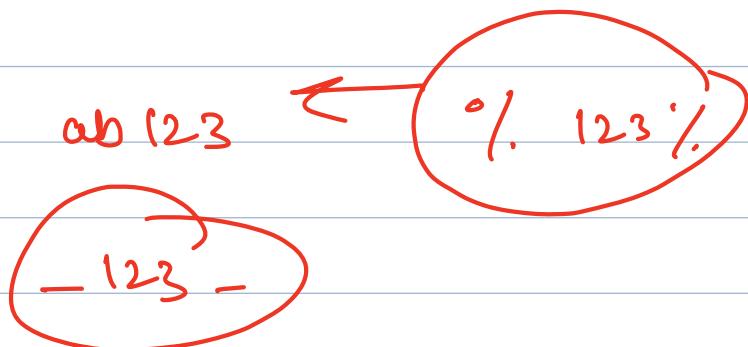
Q for a batch check if an academy batch running in morning.

name like '01 Academy 1 morning 1'

Aug 22 academy beg morning

May 22 morning academy beg
anything → academy → anything → morning
anything

select * from students
where batch-name like '%.morning%'
and batch-name like '%.academy%';



Break till 08:35 AM.

IS NULL operator

You can't use = operator to check null.

film		
id	title	description
		can be null

Q get the films that have description = null.

Select * from film
where description = NULL X

nothing = NULL

null != NULL
= =

'1' = null X

'2' == null X

IS NULL

IS NOT NULL

Select * from film
where description IS NULL ;

Students

	id	name	b_id
1	vijwal		1
2	nivek		2
3	bhraath	NULL	

Get all students
that dont belong
to b_id = 2

select * from students
where b_id != 2
and b_id is not null

ujjwal
bhavna

ORDER BY clause

→ by default order of rows in the ans is not guaranteed.

* If you want the final answer to be sorted by something, you use the

order BY clause.

select * from film
order by title_↓
desc

⇒ By default
the order
is ascending

select * from title
order by release-year, titl

tie breaker

Students

Yob	name
1998	C
1999	D
1997	C
1998	A

yob, name

1997, C

1998, A

1998, C

1999, D

Q order by year in desc order by names
in alphabetical order (ascending)

order by yob desc, title

1999, D

1998, A

1998, C

1997, C

order by job desc, title desc

1999 D

1998 C

1998 A

1997 C

student

PK

roll no.	name	phone	marks	yob
1				
2				

* In SQL default tie breaker is PK.

Code of order by:

table_name = [f }, \$S _ - - { }]
ans = []

for row in table_name:
if condition matches
ans.add(row)

ans.sort(sorting-condition)

for row in ans:
print(ans[title])

Select title from film
order by release-year.

* We can even sort on columns that we don't print.

student

id	name	yob
1	Abhi	1995
2	Abhi	1990
3	Bhem	1992

select name from

student

order by yob;

↓

Bhem

↑

Abhi

Abhi

Bhem

select distinct name

from student

order by yob;

* Can only order by coln present in
select statement \Rightarrow in case of DISTINCT

id	name	yob
1	Abhi	1995
2	Abhi	1990
3	Bhem	1992

select distinct(name, yob)
from student
order by yob.

(Abhi | 1990)
Bhem | 1992
(Abhi | 1995)

LIMIT .

Select * from film;



10000 records

* limit allows to return only a few of the results that are there from the query.

Select * from film
order by title
limit 10;

Q leave first 10 records & then show
next 20 records;

Select * from film
order by title
limit 20 offset 10;


limit 100 skip 60