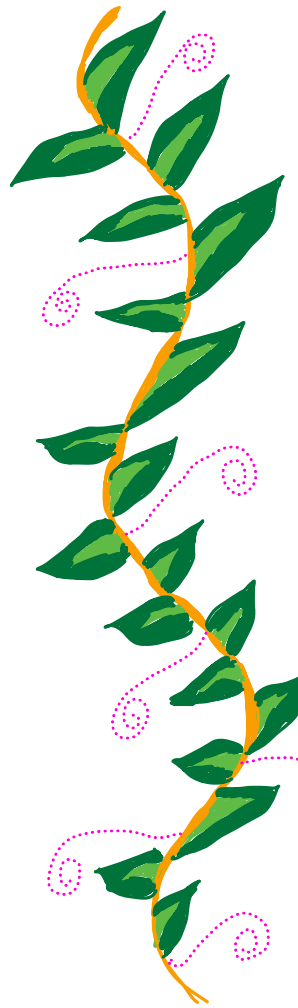


* Agenda :

- CRUD Operations
- Create
- Read
 - a) Distinct } Today -
 - b) Where
 - c) AND, NOT, OR
 - d) IN, BETWEEN
 - e) IS NULL
 - f) Order by
 - g) LIMIT
- Update
- Delete
 - ↳ Delete v/s Truncate v/s Drop



Challenge : - Complete all Assignment & do setup
tomorrow : Complete HW, clear backlog (sessions, questⁿ)



1. Read bank balance.
2. How many movies have rating > 2 .

Create :

→ Create new database	}	Create table/database name
→ Create new tables		
→ Adding new entries	→	Insert

Insert →

```
Insert into table_name  
(col1, col2, col3)  
values (val1, val2, val3);
```

Column names are optional:

→ Then all values will be mapped according to schema sequence of that table

Drawbacks of not specifying col^m names:

- 1) This is not a good practice, prone to errors.
- 2) It will be tedious.
- 3) We might have to specify values for every col^m.

Read :

select → print statement

→ Read data (full) from a table

```
select *  
from table_name
```

↳ Print full data from table-name

```
table_name = [ [ ], [ ], [ ] ]
```

```
ans = [ ] → Intermediary ans
```

```
for row in table_name:
```

```
    ans.add(row)
```

```
for row in ans:
```

```
    print(row)
```



#

Distinct :

1. Distinct should be the very first word after select
2. We can apply it on pairs as well.

Example :

```
select distinct rating , year  
from film ;
```

Pseudo Code :

```
table_name = [ [ ], [ ], [ ] ]
```

```
ans = [ ] → Intermediating ans
```

```
for row in table_name:
```

```
    ans.add(row)
```

```
filtered_ans = [ ]
```

```
for row in ans:
```

```
    filtered_ans.add ( row[rating], row[year] )
```

```
print (set (filtered_ans))
```

```
[ (2006, 'PG'),  
  (2006, 'PG') ]
```

Writing data into new table using select :

→ create table film_copy col1, col2 ...

→ Insert into film_copy
 (col1, col2)
 values () ;

```
select col1, col2  
from film;
```

Where clause :

→ It is used to filter data based upon some conditions.

→ Where clause is always used after from.

```
table_name = [ [], [], [] ]
```

```
ans = [] → Intermediary ans
```

```
for row in table_name:
```

```
    if row.matches(condition in where clause)
```

```
        ans.add(row)
```

```
filtered_ans = []
```

```
for row in ans:
```

```
    filtered_ans.add ( row[rating], row[title] )
```

```
print ( set ( filtered_ans ) )
```

Example :

```
select title, rating
from film
where rating = 'PG-13' ;
```