MySQL

1. Group By
2. Joins
3. Views
4. Indexes
5. Normalization

Group By

The `GROUP BY` statement is often used with aggregate functions (`COUNT()`, `MAX()`, `MIN()`, `SUM()`, `AVG()`) to group the result-set by one or more columns.

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

Example

dept_id	dept_name
1	DS
2	Developer
NULL	NULL

emp_id	emp_name	dept	salary
1	Rahul	2	30000.00
2	Mayank	1	36000.00
3	Ajay D	1	45000.00
4	UshaKumari	2	42000.00
5	Swapnil	1	60000.00
6	Chetan	1	75000.00
7	Kiran	1	65000.00

```
select count(e.dept), d.dept_name from emp_data e, dept_data d
where e.dept = d.dept_id
group by dept;
```

Joins

A `JOIN` clause is used to combine rows from two or more tables, based on a related column between them

```
 7
 8 •  select * from emp_data e
 9    INNER JOIN emp_details d ON d.id = e.emp_id;
10
11
12
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

emp_id	emp_name	dept	salary	ID	emp_name	location
1	Rahul	2	30000.00	1	Rahul	Banglore
2	Mayank	1	36000.00	2	Mayank	Mumbai

LEFT JOIN

```
8 •  select * from emp_data e
9    LEFT JOIN emp_details d ON e.emp_id = d.id;
10
11
12
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

emp_id	emp_name	dept	salary	ID	emp_name	location
1	Rahul	2	30000.00	1	Rahul	Banglore
2	Mayank	1	36000.00	2	Mayank	Mumbai
3	Ajay D	1	45000.00	NULL	NULL	NULL
4	UshaKumari	2	42000.00	NULL	NULL	NULL
5	Swapnil	1	60000.00	NULL	NULL	NULL
6	Chetan	1	75000.00	NULL	NULL	NULL
7	Kiran	1	65000.00	NULL	NULL	NULL

Right JOIN

```
/
select * from emp_data e
RIGHT JOIN emp_details d ON e.emp_id = d.id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

emp_id	emp_name	dept	salary	ID	emp_name	location
2	Mayank	1	36000.00	2	Mayank	Mumbai
1	Rahul	2	30000.00	1	Rahul	Banglore
NULL	NULL	NULL	NULL	12	M S Dhoni	Chennai

Views

```
CREATE OR REPLACE VIEW view_name
AS
SELECT column1, column2, ...
FROM table name
WHERE condition;
```

```
DROP VIEW view_name;
```

Indexes

Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.

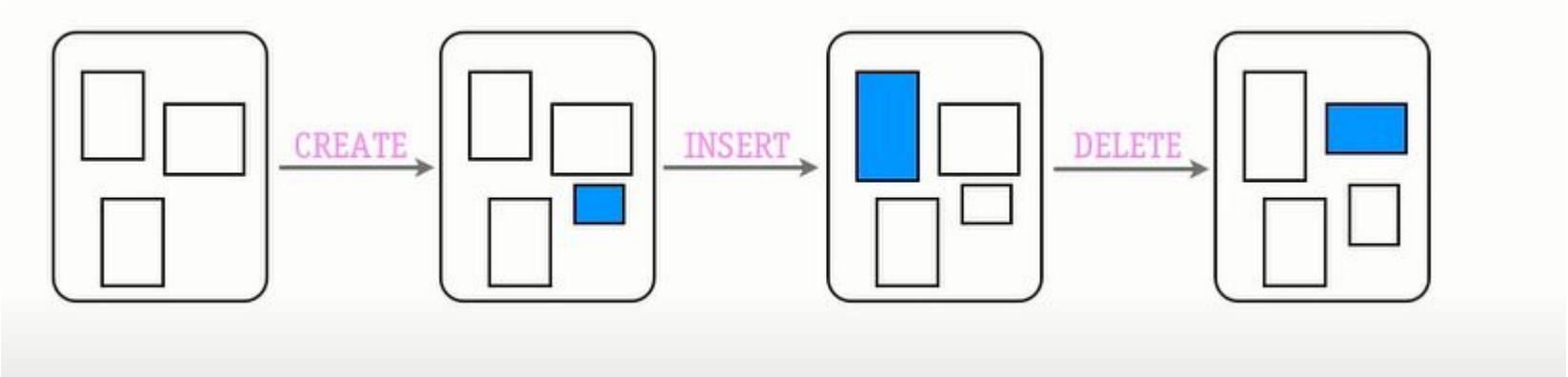
CREATE INDEX Syntax

Creates an index on a table. Duplicate values are allowed:

```
CREATE INDEX index_name
ON table_name (column1, column2, ...);
```

```
ALTER TABLE table_name
DROP INDEX index_name;
```

Transaction in DBMS



Bank Account

acctnum	balance
100	504.92
105	1250.24
110	750.31

SQL Commands

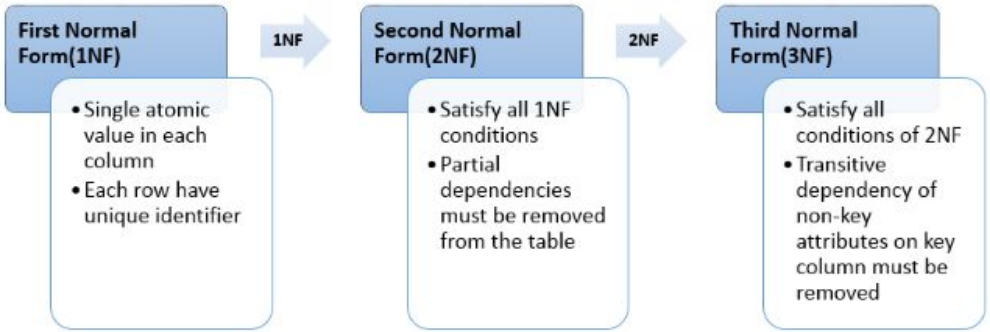
```
BEGIN;

COMMIT;

ROLLBACK;
```

Normal forms at a glance

Its time to summarize our reading. We have below image to summarize the reading on normal forms:



Database normalization