

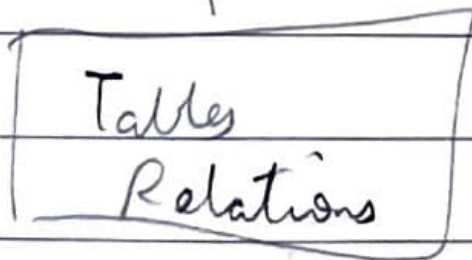
Lec 52: Introduction to Structured Query Language

User



Language \rightarrow SQL

Database



In 1970, EF Codd published a paper on Relational Model. He took the help of relational

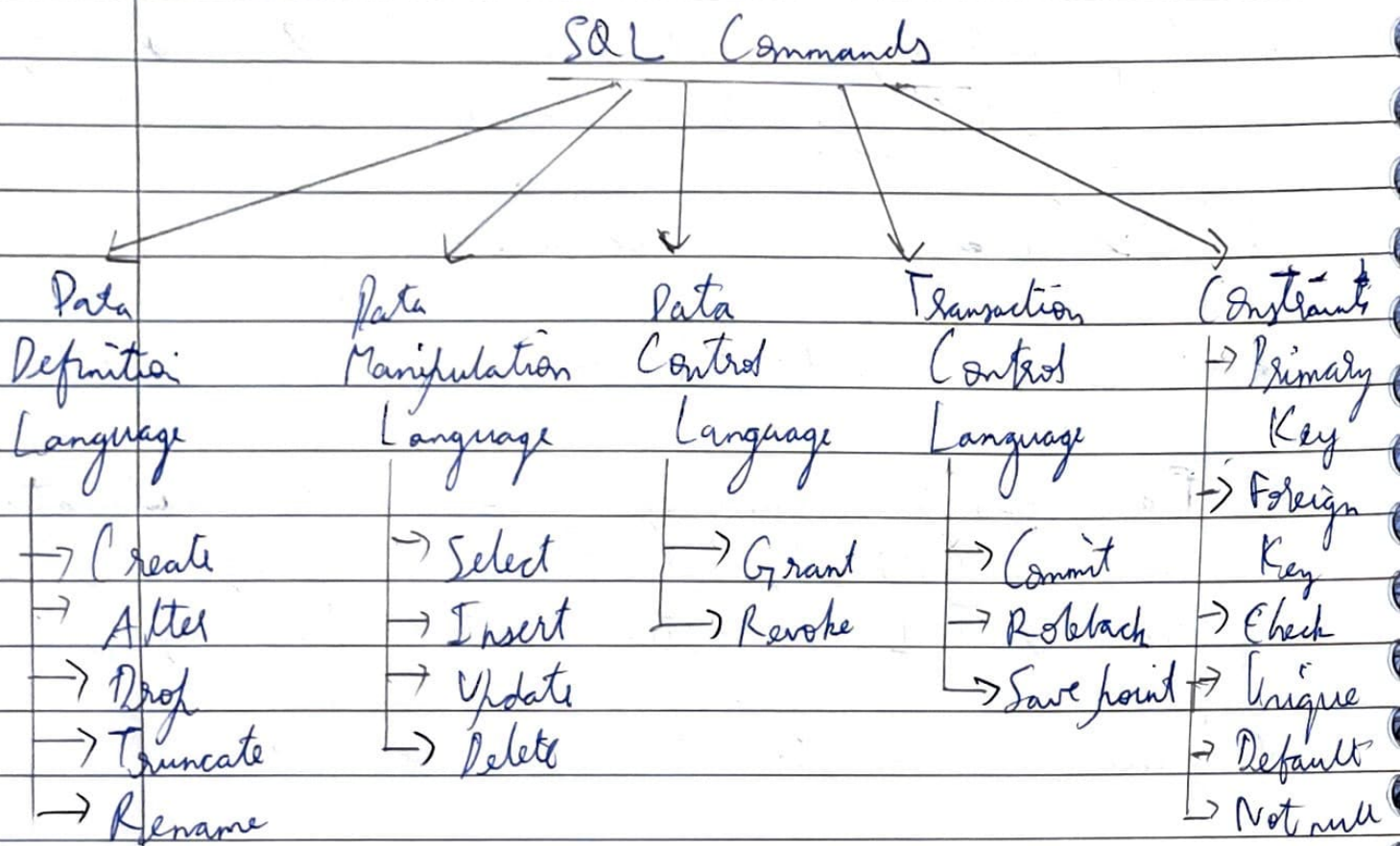
& tuple relational calculus

—1—

algebra, to prove that data can be stored

- * IBM ~~convent~~ implemented this research findings by developing Simple English Query Language [SEQUEL], which became SQL.
- * SQL is a domain-specific language
- * SQL is a declarative language
- * Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL) and TCL (Transaction Control Language).
- * Keys and constraints Exist/Not exist
- * Operators (Like, between, In, Not in, Conditional)
- * Clauses (distinct, order by, group by, from, having)
- * Aggregate functions
- * Joins and nested queries ***
- * PL SQL (Triggers, function, cursor, procedures)
- * Declarative \rightarrow What ^{to} do
&
Procedural \rightarrow What ~~for~~ to do, How to do

Lec-53: All Types of SQL Commands with Example



Lec-54: Create table in SQL with execution

```
create table <table_name>
(
    column1 name datatype,
    column2 name datatype,
    column3 name datatype,
);
```

```
desc table_name;
```

Eg:- create table emp

```
(
    id int,
```

```
name varchar2(20)  
salary number(10)  
);  
desc emp;
```

~~TABLE~~

Lec-55: ALTER Command (DDL) in SQL

- Add column/s
- Remove column/s
- Modify datatype
- Modify datatype length
- Add Constraints
- Remove constraints
- Remove column/table

To add a ~~col~~ column to a table

Alter ~~table~~ Student Add { address varchar(20);

To drop a column from the table.

alter table employee drop column address;

To modify the datatype of ~~the~~ ~~table~~ a column

alter table employee modify id varchar(10);

If you want to change the length of varchar,

alter table employee modify id varchar(30);

To rename a column name

alter table employee rename column id to roll-no;

To rename the table name

alter table employee rename to emp11;

To add primary key to the table

alter table emp11 add primary key (roll-no)

Lec 56: Difference between Alter & Update in SQL
with examples

	Alter	Update
*	DDL	DML
*	Used to change the schema.	Used to change the data.
update	Emp set Salary = Salary * 2 where id = 1;	

Lec 57: Difference between Delete, Drop & Truncate in SQL

	Delete	Drop	Truncate
*	DML	DDL	DDL
*	Delete from table name Delete from Student (Rows will be deleted)	Drop table Student; (The entire scheme will be deleted)	Truncate Student; (Rows will be deleted)
*	There is a where option		There is no where option
E.g.	Delete from student where id = 1;		
*	Delete is slow		Truncate is faster
*	There is rollback option		There is no rollback option

Lec 58: Constraints in SQL

1) Unique

2) Not Null → The value should not be empty.

3) Primary Key = Unique + Not Null.

4) Check

Eg: Check (age > 18)

5) * Foreign Key

6) Default

Eg: Salary int default 10000

Lec-59: SQL Queries & subqueries (Part-1)

1) Write a SQL Query to display max salary from emp table

2) Write a SQL Query to display employee name who is taking maximum salary.

Emp

E_id	E_name	Dept	Salary
1	Ram	HR	10000
2	Amit	MRKT	20000
3	Ravi	HR	30000
4	Nitin	MRKT	40000
5	Varun	IT	50000

Max

Soln: 1) Select (Salary) From Emp;
~~Where Max~~

2) Select E_name From Emp
Where Max(Salary)

2) Select E_name from Emp
Where Salary = (Select Max(Salary) from Emp);

Lec-601 - SQL Queries and Subqueries (Part-2)

3) Write a SQL Query to display Second Highest Salary from Emp table

4) Write a SQL Query to display Employee name who is taking second highest salary.

Soln 3) ~~Select Max(Salary) from Emp~~
~~Max~~ ~~!=~~
Select Salary from Emp Where Salary ~~!=~~ (Select Max(Salary) from Emp);

4) Select ~~Emp~~ E_name from Emp
Where Salary = (Select Max(Salary) from Emp
Where Salary != (Select Max(Salary)
From Emp));

Lec-611 - SQL Queries and Subqueries (Part-3)

5) Write a query to display all the dept names along with no. of emps working in that.

Soln. ~~Select Dept~~ Count(Dept) from Emp

Select dept, Count(Dept) from Emp
Group By dept;
GROUP BY

Lec 62: SQL Queries and Subqueries (Part-4)

6) Write a query to display all the dept names where no. of emps are less than 2.

Soln. ~~SELECT DEPT FROM EMP~~
~~WHERE COUNT(DEPT)~~

~~SELECT DEPT~~

select Dept from Emp group by Dept
having count(*) < 2;

7) Write a query to display the ^{emp name} employee name ~~from the~~ belonging to the dept where no. of employees are less than 2.

Soln. select E_name from Emp
where Dept in (
select Dept from Emp group by Dept
having count(*) < 2;
);

Lec 63: SQL Queries and Subqueries (Part-5)

8) Write a query to display highest salary department-wise and name of the employee whose is taking that salary.

~~But~~ select E_name, Salary from Emp
 where Salary in (
 select max(Salary) from Emp group by Dept);

Lec 64: SQL Queries and Subqueries (Part-6)

Use of IN and not IN

Emp			Project				
<u>Eid</u>	Ename	Address		<u>Eid</u>	<u>Pid</u>	Pname	Location
1	Ravi	Chd		1	P ₁	IoT	Bangalore
2	Varun	Delhi		5	P ₂	Big Data	Delhi
3	Nitin	Pune		3	P ₃	React	Mumbai
4	Robin	Bangalore		4	P ₄	Android	Hyderabad
5	Ammy	Chd					

Query to get the details of employees whose address is either Delhi or Chd or Pune

Soln. select * from Emp
 where ~~Address~~ Address in ('Delhi', 'Chd', 'Pune')

Lec 65: SQL Queries and Subqueries (Part-7)

Use of IN and not IN

Find The ~~number~~ name of Emp who are working on a ~~time~~ project

~~select~~ select * from Empname from ~~Emp~~ Emp
natural join Project

(OR)

```
select Empname from Emp
where Empid in (
    select Empid from Project)
```

Lec-66: EXIST and NOT EXIST subqueries (Part-8)

Find the details of Emp ~~who~~ who is working on at least one project.

```
select * from Emp
where exists exists (
    select Empid from Project where Empid = Project - Empid)
```

True

Nested query is bottom-up approach
Correlated query is top-down approach