

Dec. 2011 / Jan. 2012



Saturday 31

365 - 000 / Week 52

## Database SQL

Database is a collection of information  
Organized for easy access, management  
and maintenance.

### DBMS operations:-

- Adding new files
- Inserting data
- Retrieving data
- Modifying data
- Removing data
- Removing file.

### Normalization:-

→ Decompose larger, complex table into simpler  
and smaller ones.

Sunday 01

001 - 365 / Week 52

→ Move from lower normal forms  
to higher normal forms.

It reduce redundancy.

J	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T
A N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

2012

SQL

Structured Query Language:-

SQL Data Types

1. Numeric :- bit, tinyint, smallint, int, bigint, decimal, numeric, float, real
2. Character/String - char, varchar, text.
3. Date/Time - Date, Time, Datetime, Time Stamp, Year.
4. Miscellaneous - Json, XML.

Constraint

- 1) Not null
- 2) Default
- 3) Unique
- 4) Primary
- 5) Check
- 6) Index

J	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T							
A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
N																															

2012

J	S	M
A	N	1 2

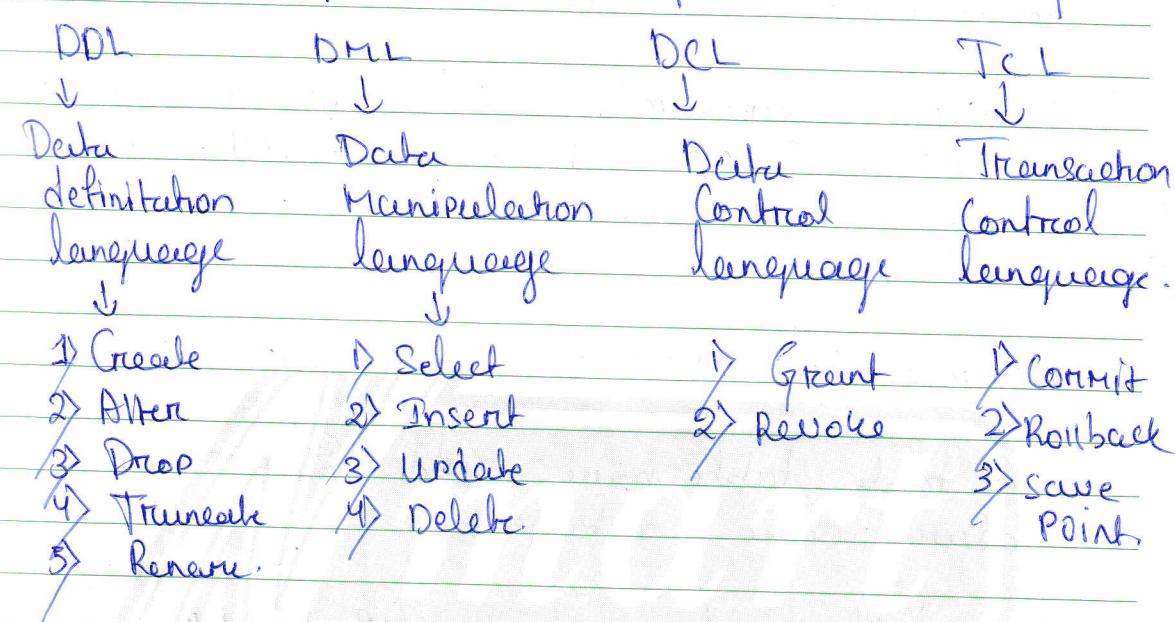
# January



Tuesday 03

003 - 363 / Week 01

SQL



DQL

↓  
select.

J	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T
AN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

2012

T  
31  
  
2012

→ Create Command :-

Create table table\_name ( Emp\_id INT (10) Not Null,  
 First\_name Varchar(20), last\_name Varchar(20)  
 Not Null, Salary int (10) Not Null,  
 Primary key (Emp\_id));

→ Alter Table Command :-

→ Alter table table\_name add ~~Column~~  
 Column\_name datatype;

→ Alter table table\_name Rename Column

Column\_name to new Column\_name;

→ Truncate Command :-

Truncate table table\_name;

→ Drop Command :-

drop table table\_name;

J	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T							
A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
N																															

2012

# January



Thursday 05

005 - 361 / Week 01

DML

Data manipulation language.

- Insert
- ii) Update
- iii) Delete

Insert Command :-

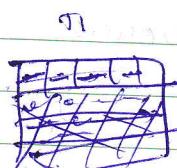
Insert into employees (Emp\_id, first\_name,  
last\_name, salary) Values (101, 'steven',  
'king', 10000);

Update :-

→ Update Stud Set name = 'Avisek' Where Std\_id = 101.

Delete :-

→ Delete from Employees Where Emp\_id ~~= 101~~  
(101, 103)



J	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T
JAN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

2012

# January

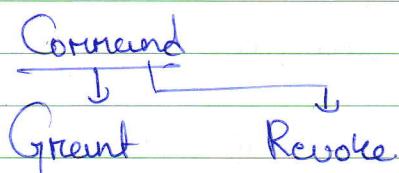


Friday

06

006 - 360 / Week 01

## DCL Data Control Language



Description

## TCL Transaction Control Language

Command

- Commit
- Rollback
- Savepoint

Where clause:

Select \* from Employees Where Emp.Id = 101;

And operator

Select \* from Employees Where Emp.Id = 101  
and Salary >= 10000;

J	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T							
JAN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

2012

J	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T							
JAN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

06

Week 01

# January



Saturday 07

007 - 359 / Week 01

## Special Operators

Between

Like

IS NULL

IN

DISTINCT

&gt;

&lt;=

&lt;

&gt;=

&lt;&gt; or !=

## Comparison operators

→ Select \* from employees where salary between 1000 and 2000;

→ Select \* from employees where first name like 'steven';  
 ↳ (first-name starting from s).

## SQL Operators:-

(Y. Ø)

Y. Ø%

## Aggregations

Avg()

Select Avg(Salary) from employees;

Sunday

08

Count()

Select Count(\*) from employees;

Max()

Select Max(Salary) from employees.

Min()

Sum()

1015

101

S	S	M	T	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T
28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

2012

008 - 358 / Week 01

JAN	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

2012

# January



Monday 09

009 - 357 / Week 02

## Group By clause:

Arrange identical data into groups.

Select Max (Salary), dept\_id.  
from Employees.  
Group by dept\_id.

## HAVING clause

Opposite with aggregate clause

Where clause not used :-

Select avg(Salary), dept\_id.  
From Employees group by  
dept\_id having count(dept\_id) = 2.

## Order by clause

Used to sort output :-

Select \* from Employees Order  
by salary desc or asc)

J	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T
JAN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

2012



## UNION:

Used to combine the result-set of two or more select statement removing duplicates.

→ Select name from Stud Union Select first\_name from Employees.

## SQL JOINS

If is used to combine two or more than two table/relation.

### Types of JOIN

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN

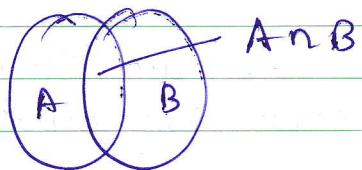
- Returns rows where there is a match in both tables.

S	M	T
29	30	31
30	31	

S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
29	30	31																						

## Inner JOIN

Returning rows when there is a match in both tables.



## LEFT JOIN

Returning all rows from the left table,

Even if there are no matches in the right table

$$(A \cap B) \cup A$$

## Right JOIN

Returning all rows from Right table

Even if there are no matches in the left table.

and return rows of left table if it matches.

J	S	M	T	W	T	F	S	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T						
JAN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

2012

J	S	M	T	W	T	F	S	S	M	T
JAN	1	2	3	4	5	6	7	8	9	10

# January



Thursday 12

012 - 354 / Week 02

## Full outer JOIN :-

Returning Rows where there is a Match in one of the tables

## Self JOIN :-

Used to join a table to itself as if the table were two tables, temporarily renaming atleast one table in the SQL Statement

## Cartesian JOIN (Cross JOIN) :-

Returning the Cartesian Product of the sets of records from the two or more joined tables.

Cross Product.

## Inner JOIN

Select table1.col1, table1.col2, ..., table1.coln.  
From table1

Inner JOIN table2.

ON table1.common col = table2.common col;

S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

2012

# January



Friday

13

013 - 353 / Week 02

→ Select e.emp\_id, e.first\_name, e.last\_name,  
d.dept\_id from Employees.e  
InnerJoin on e.dept\_id = d.dept\_id;

↓  
department\_d

④ ↓  
→ Same Syntax for left and Right Join.

Select table1.col1, table1.col2, table1.coln

From table1 LeftJoin table2 ON

table1.commonfield = table2.commonfield;

Full Outer Join:

Left

LeftJoin

+

Union

+

RightJoin

J	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	
A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
N																									31

2012

# January

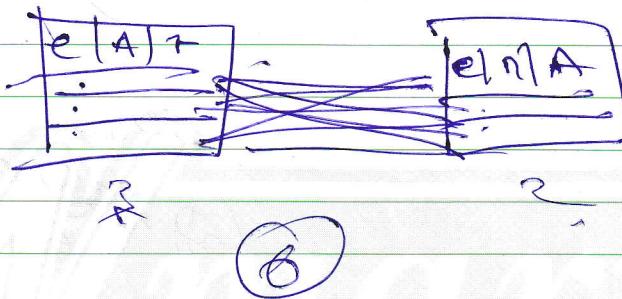


Saturday 14

014 - 352 / Week 02

Cross JOIN

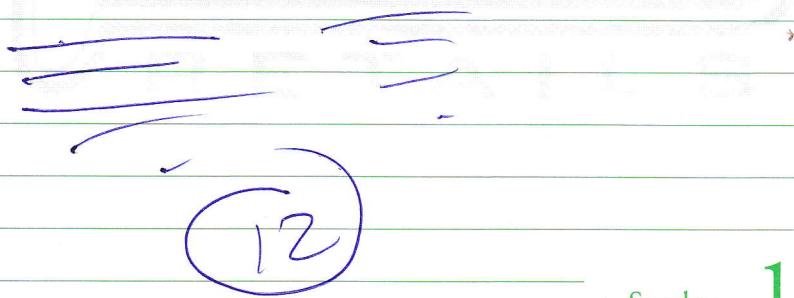
Select \* From table1 cross join table2;



(8)

Sunday 15

015 - 351 / Week 02



(12)

M	T
30	31
2012	

	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

2012