

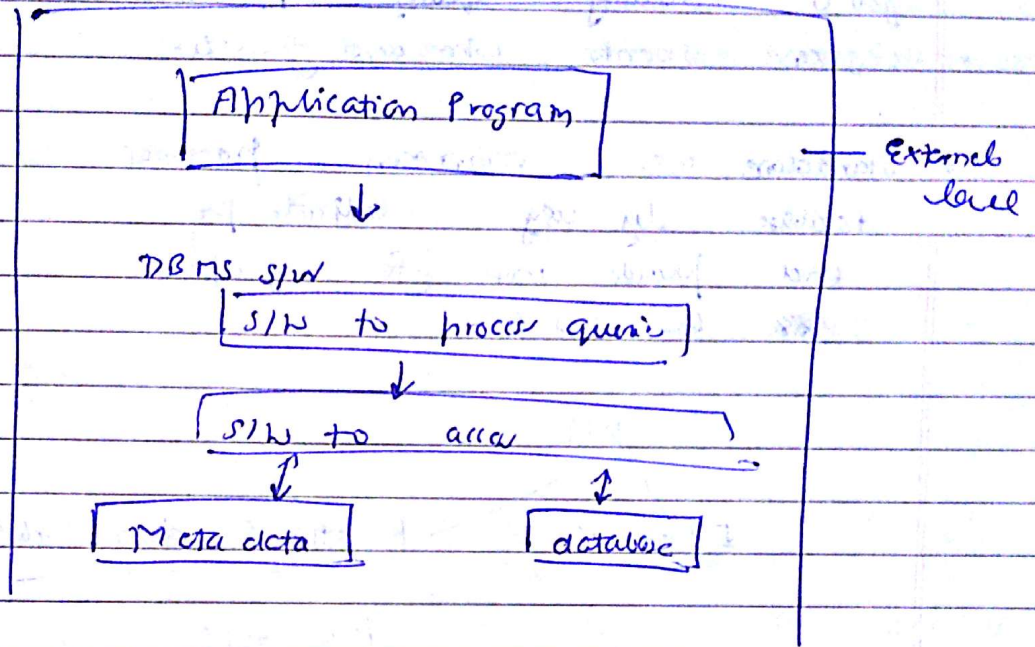
Book Corlh

Practical - Van Bayer



DATE: ___/___/___
PAGE: ___

DBMS



→ DB administration

It ~~not~~ include

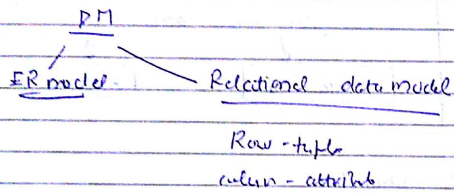
- (a) Schema definition - it create original database schema by executing DDL statements
- (b) Storage structure & access method definition -
- (c) Grant of authorization for data access -
- (d) Routine maintenance - Periodically backup up database either on tape or remote server
- (e) Monitor job running on database and insure that performance is not degraded by expensive task submitted by user
- (f) Be responsible for identifying data to be stored

→ occasionally access database and need different information everytime.

GOOD WRITE

→ naïve for parameter users
job is constantly query updation
engine scientist who are familiar

• Standalone user - maximum personnel
database by using dedicated for
and provide more for graphics
base interface.



Other data models

① Object oriented data model = study features
like encapsulation etc with ER model.

② Object relational data model
combines features of object oriented data
model and relational data model.

2 obsolete data models

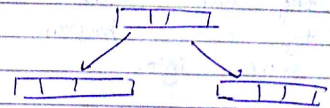
① Network data model =



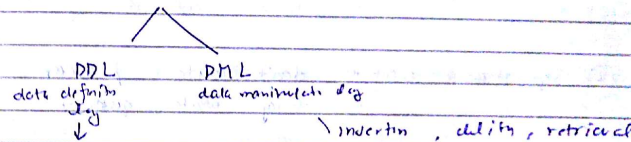
GOOD WRITE

② Hierarchical data model =

data in form of records in form of tree structure



Database languages =



- DDL - database schema is specified by a
set of definitions expressed by a
special language called data definition
language.

- DML -

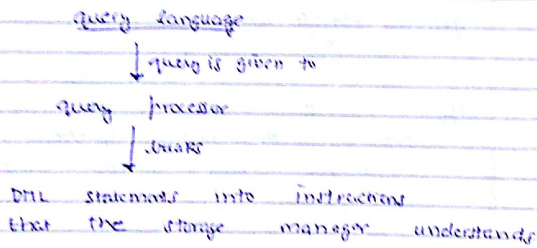
② Procedural Non procedural

① Procedural - is DML to which user can specify what data is needed &
how to get that data.

GOOD WRITE

Database users:

- ① Naive users - unsophisticated users
 - they don't interact with system
 - they make program to retrieve info.
- ② Application programmers - interact with system by writing application programs.
- ③ Sophisticated user - don't write program - they write queries



- ④ Specialised users - interact by writing database applications
 - API - CAD
 - Knowledge base system

DBA - database administrator

- A person having centralised control over the systems

Roles:

- ① Schema definition - by using statements in DDL
- ② Storage structure and access method definition
- ③ Schema and physical organisation modification
- ④ Branching of authorisation for data access - security
- ⑤ Routine maintenance
 - back up
 - enough disk space available
 - monitoring day to day jobs

SQL

- Commercial database mg. a lang. is that is user friendly, it SQL is more influential market for query lang. languages
- SQL is a non procedural
- SQL is query lang. but it defines structure of data, modified data & strictly security constraint

① DDL

It provides command for defining relational schema, deleting and modifying relation schema.

① Create - used to create table or database

SQL Name
Name varchar(20)
P.S

② Drop -

③ Truncate -

④ Truncate - remove all rows from table
it cannot be rolled back or triggered.
Truncate - of Table S

- Truncate Table Student

⑤ Rename -

Rename Student

⑥ Alter command -

Modify structure of existing table

- Add a new column
Alter table Student ADD (Age number(2));

GOOD WRITE

- Alter table drop column

- Alter table student modify (Name, Varchar(20))

② DML - one database is populated with data
there is need to perform manipulation

- manipulation mainly retrieval, insertion, deletion & modification of data

① Select

② Insert - to insert data

③ Update - changes to data

④ Delete - to delete all rows from database.

③ DCL Data control language

It is used to provide control access to data stored in database.

① Grant - to allow specific user to perform specific task

② Revoke - to deny permission

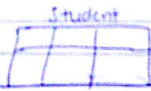
④ TCL - Transaction control language
- It manage change made by DML statements

GOOD WRITE

Commit, rollback, checkpoint, set transaction

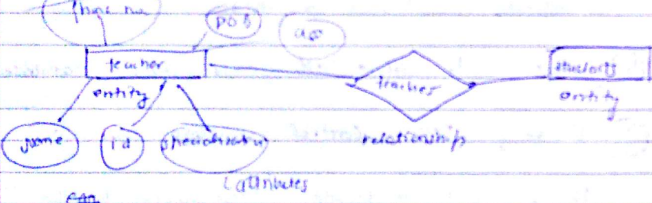
Basic structure of SQL query

- 1) Select -
- 2) From -
- 3) Where -



→ Select Name from Student where <condition>

ER model - entity relationship model.



Types of attributes -

- 1) Single and composite attribute
- 2) Single valued and multivalued attribute
- 3) Derived attributes
eg. age is derived from DOB

Binary relationship - 2 element sets
Ternary relation - 3 element sets

GOOD WRITE

- Buffer
- date dictionary
- used of DBMS

String operator (like operator)
if in SQL we write strings in ' ' strings marked are case sensitive

- like operator is used for string comparison with the help of
 - a) Percentage (%) - %a - last letter a | a% - first letter a
 - b) Underscore _ - 0 or any no of string makes only one character

eg. Find name of all customer whose name end with a

Select name from <table name> where name = name like ' %a';

eg. Customer start 2nd letter is O.

Select name from Customer where name like ' _o %';

Comparison with NULL - A special comparison operator is

NULL IS NULL / IS NOT NULL is used to test a column for NULL values.

eg. Find acc no from account table where balance is NULL

Select acc no from account where balance is NULL

GOOD WRITE

order by tuple-



SQL allow user to specify order in which tuple are displayed why default order is ascending & if user want to display the data in descends order then DESC clause is used order by [column1] [column2] desc;

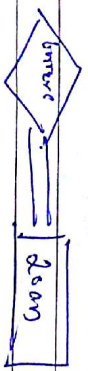
Q Find customer in alphabetical order who are having the born at month XY?2

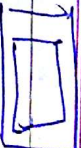
SELECT customer from customers where bornname as B, born as 1 where B.bornname = 'X.Y.Z' and B.name = 'X.Y.Z' order by cname;

Binary set operators:-

Symbolic in ER diagrams.

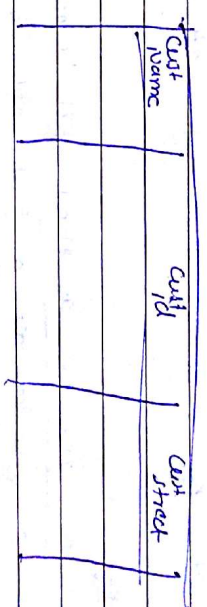
- Double ellipses - represent multi valued attribute
eg. Phaseno 
- Dashed ellipses - represent derived attribute 
- Double line - represent total participation of an entity in a relationship set



→ Double rectangles -  represent weak entity sets

Key - a key is a set of attributes or a single attribute that helps to distinguish entities from each other

→ Super key - a set of one or more attributes that collectively identify uniquely an entity in the entity set.



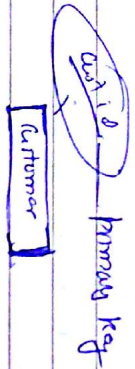
→ Customer + Custstreet - Super key
then which shouldn't form a unique entity

→ Candidate key - set of attributes that uniquely identify an entity

37 Primary key - There are no of candidate keys for an entity set. The candidate key that the database designer chooses to identify an entity. Usually the primary key is chosen with care that it should not change with

GOOD WRITE time.

→ Alternate Key - Those are the no. of candidate keys in a table. The candidate keys that are not chosen as primary keys are called as alternate keys.



Candidate - one to one, one to many, many to one, many to many

Diff b/w Primary and Super Key -

Super Key - may have extra attributes
 candidate key do not have extra attributes

Candidate Key -

customer name + cust street, the subjects are
 cust name, cust street we use that name of the subject is a super key
 with minimal super key are called candidate key

Super Key - cust id is a super key

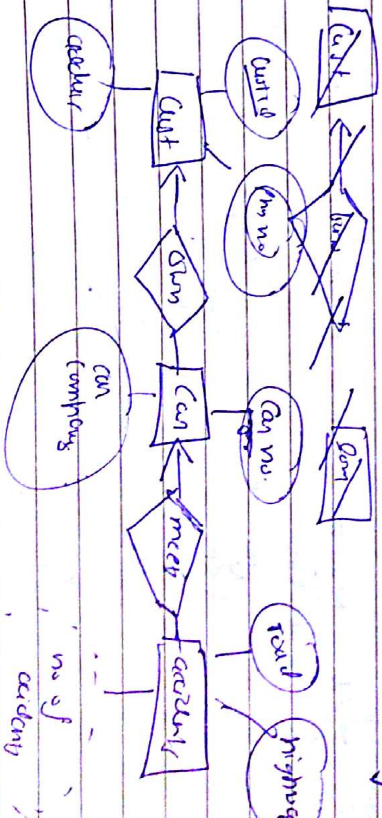
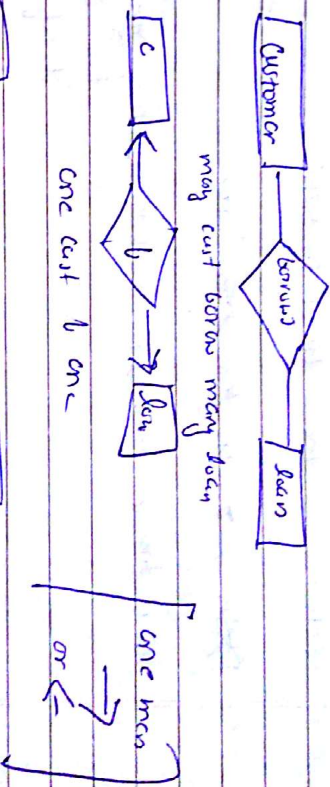
• cust name + cust id - Super Key
 super key may have extra attributes

o determine attribute -

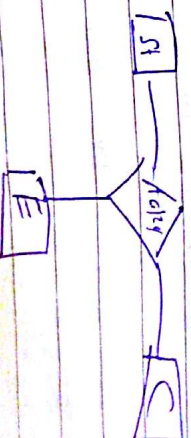
GOOD WRITE

→ ER diag

Construct an ER diag for a car insurance company where customer own one or more cars each. Each car can have associated with it 0 to any no. of recorded accidents.



o Construct an ER diag that model student as one entity, set, result of 2nd, exam of 3rd



GOODS WRITE

Nested Subqueries -

are used in where and having clause

Q7 IN Subquery (IN/ = Any / = Same) [abstract]
SQL allows > / < / >= / <= operator with Any and SOME.

Q8 Find name of all branches that have assets and atleast one branch in Delhi.

~~Select~~ ~~branch~~ ~~from~~ ~~branches~~ ~~where~~

Select branchname from branches where exists (select asset from branches where branchname = r.branchname)

All Registered must match all rows in the subqueries

when you use some or any

Delhi	Delhi
100	100
200	200
300	300
400	
500	

UP(400, 500) / by any row or any we get 200, 300, 400, 500 as output
when we use all we get /400, 500 as output

Find name of all branches that have assets

all branches located in Delhi.

select branchname from branches where exists

GOOD WRITE

for highest value

Find branch that have highest avg values -

Select branchname from accounts ~~having~~ ~~max~~ ~~avg~~ ~~branch~~ ~~by~~ ~~branchname~~ ~~having~~ ~~max~~ ~~avg~~ ~~(balance)~~ ~~>~~ ~~all~~ ~~(select~~ ~~max~~ ~~avg~~ ~~(balance)~~ ~~from~~ ~~accounts)~~

- implicit cursor (automatically created when SQL is executed)
- explicit cursor

how will update the table & row salary of each employee by 500 & use the SQL %
% count = % NET FUND

% 10 OPEN
% 20 PENDING

open the cursor

OPEN C-emp

Fetch the cursor

Fetch C-emp INTO C-rid, C-salary, C-oth;

close C-emp;

GOOD WRITE

10 Saturday

JANUARY '15

	M	T	W	T	F	S	S	
			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							

DBMS - Unit 1, 2

→ Relational algebra operations -

1. Select (σ) - Row
2. Project (π) - Column
3. Union (∪)
4. Set difference (-)
5. Cartesian Product (X)
6. Rename (ρ)

SQL

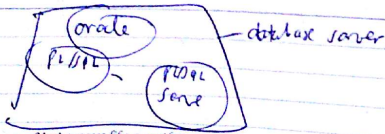
PL/SQL - program logic

- | | |
|--|---|
| <ul style="list-style-type: none"> - Create table - manage data - Control data - Transaction | <ul style="list-style-type: none"> - Conditional instruction (if then etc) - loops - Exception handling - Stored routines |
|--|---|

- Oracle uses PL/SQL engine to process PL/SQL statements

→ PL/SQL language code can be stored in client system or in database

Priorities

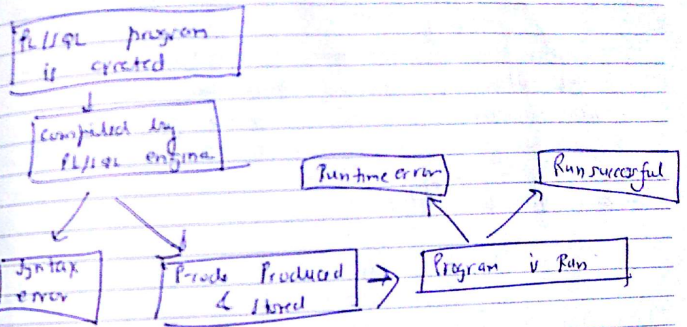


Meet success like a gentleman and disaster like a man

JANUARY '15

	M	T	W	T	F	S	S	
					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					

PL/SQL program Monday 12



PL/SQL Program Structure

DECLARE ← Declaration Section - optional

Execution section → Program logic - mandatory

BEGIN
END → Exception logic → optional

EXCEPTION

Priorities

→ DECLARE variable → cursor

BEGIN Program

EXCEPTION exception handling

The superior man makes the difficulty to be overcome his first interest; success comes only later

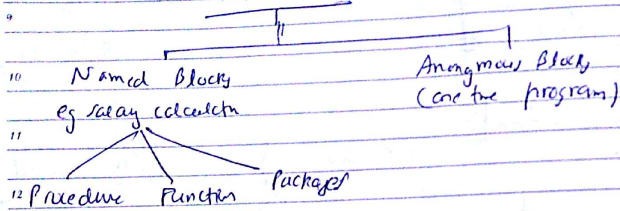
13 Tuesday

WK 03 • Day 013

JANUARY '15

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

PL/SQL Block



PL/SQL Cursor

- Used to process individual rows returned by database system for a query
- Cursor holds rows returned by SQL statement
- Cursor is defined as work area where st. is executed.

Cursor

Implicit	Explicit
- predefined	- user defined
- associated with DML st.	- associated with SELECT st. that return more than one row.
- called SQL cursor	- declared in declaration section & used in execution section
Priority	Commands used
OPEN	OPEN, FETCH, CLOSE
% NOT FOUND	

Success is the proper utilization of failure

JANUARY '15

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

Wednesday 14

WK 03 • Day 014

Data Independence

- Logical - conceptual schema can be changed without affecting existing external schema
- Physical data independence - physical storage structure or devices used for storing data could be changed without change in conceptual or external view

User

1. Native - need not aware of database system
2. Oracle - user may communicate with database directly or indirectly

DBA

3. schema architecture
 1. internal - describe physical storage structure of database.
 2. conceptual - hide detail of physical storage structure & describe entities, data type, relationship, user character.
 3. external - describes part of database that a particular user group is interested in and hides rest of database

Priority

I don't know the key to success, but the key to failure is trying to please everybody

15 Thursday

Week 03 - Day 015

JANUARY '15

M	T	W	T	F	S	S	M	T	W	T	F	S	S		
					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										

DBA

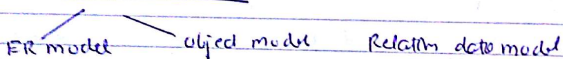
1) create original database schema by
execute DDL statements

- 2) New software installation
- 3) Security enforcement
- 4) Data backup
- 5) Maintenance check

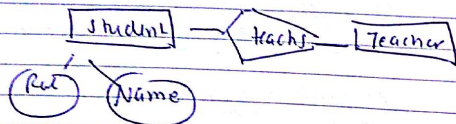
ESSDA M

MSI
SERVITS

Conceptual Model



- entities
- attributes
- relationships



entity - any distinguishable object that has an independent existence

- Priorities
- Tangible - eg book
 - Nontangible - eg knowledge

He has achieved success who has worked well, laughed often, and loved much

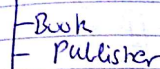
JANUARY '15

M	T	W	T	F	S	S	M	T	W	T	F	S	S
					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								

Friday 16

Week 03 - Day 016

Entity type, entity set



Attribute

- single & multivalued
- ~~simple~~ - simple & composite
- Stored derived

Key attribute - superkey, candidate key, primary key, foreign key

Notation

Purpose



Entity type



Relationship



Attribute



Key attribute

Priorities



derived attribute



multivalued attribute

... the distance you traveled from where you started

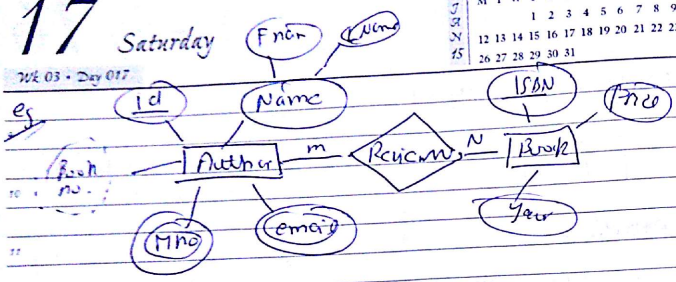
17

Saturday

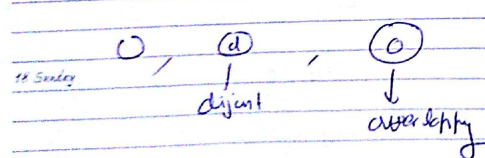
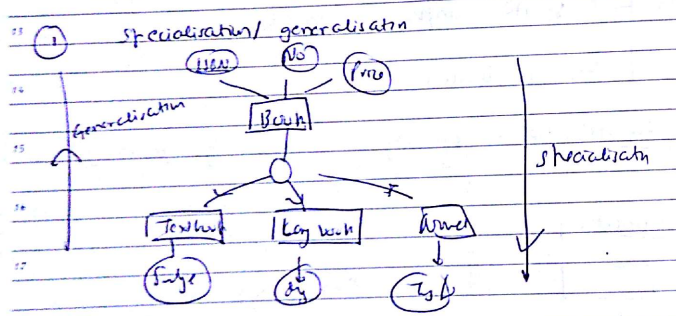
JANUARY '15

M	T	W	T	F	S	S	M	T	W	T	F	S	S
							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				

70% 03 - Day 017



Enhanced ER-model -



Priorities

The toughest thing about success is that you've got to keep on being a success

JANUARY '15

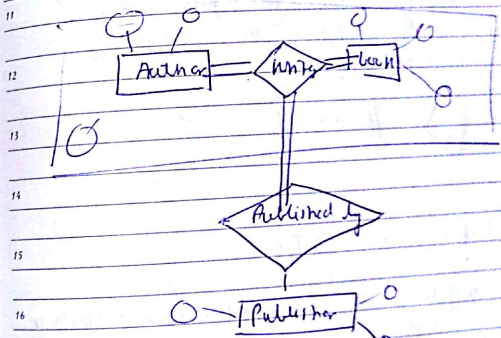
M	T	W	T	F	S	S	M	T	W	T	F	S	S
							1	2	3	4	5	6	7
							8	9	10	11	12	13	14
							15	16	17	18	19	20	21
							22	23	24	25	26	27	28

Monday 19

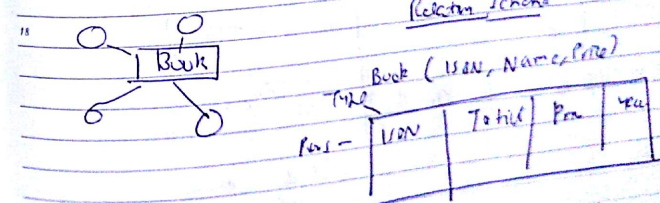
70% 04 - Day 018

Aggregation -

- process through which one can treat relationship as higher-level entities
 - It show relationship among relationship



Translating FR to relational



Priorities

Failure is the companion that gives success its flavor

20 Tuesday

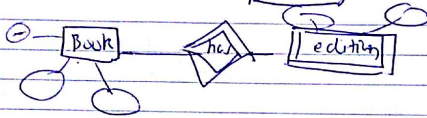
Wk 04 - Day 020

JANUARY '15

	M	T	W	T	F	S	S	M	T	W	T	F	S	S
								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								

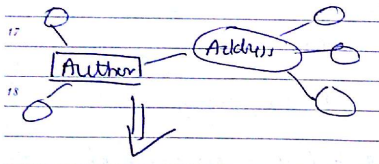
Mapping of ER model to relational model result in relation that is known as entity relation

→ Weak entity type to relation - The primary key of weak entity consist of primary key of strong entity type or foreign key and its own partial key.



EDITION = (ISBN, edition no, Type)

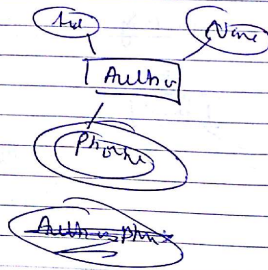
Composite Attribute



Author (Aid, Name, city, State, Post)

Priorities

Multivalued attribute



Author-Phone (Aid, Phone)

Aid	Phone

Success is the maximum utilization of the ability that you have

JANUARY '15

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

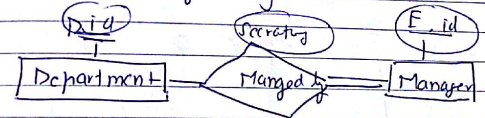
Wednesday 21

Wk 04 - Day 021

Relation Type

① one to one

- choose one of them or base relation
- choose one as base
- include primary key of other entity type as foreign key of base



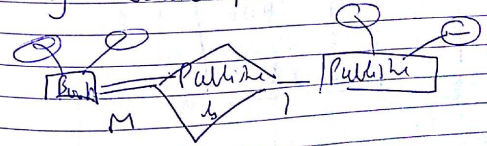
base we choose

↓ Relation

Manager (E_id, Did, Secretary)

② one to many or may to one

- primary key of entity type on many side of relationship



Priorities

Book (ISBN, Title, List date, P_id)

The ladder of success is never crowded at the top

22 Thursday

JANUARY '15

M	T	W	T	F	S	S	M	T	W	T	F	S	S	
				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									

2024-01-22

Plan to copy

Select

$\sigma(r) \{ \neg (t \in r \text{ and } p(t)) \}$

\downarrow
i-dm

eg

A	B	C
1	1	7
1	3	5
3		6

$\sigma_{A=B} \Rightarrow$

A	B	C
1	1	7

$\sigma_{A \neq B}$ isom

Project, Select and SQL

employee

Fid	Did	Salary
111	1	200
222	1	400
123	2	300
333	2	500

Project $\pi_{Fid}(\sigma_{Did=1}(employee))$

\downarrow

Fid
111
222

You always pass failure on the way to success

JANUARY 23

M	T	W	T	F	S	S	M	T	W	T	F	S	S
				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								

Friday 23

2024-01-23

Select Fid
from employee
where Did = 1 } SQL

Difference (-)

$r - s = \{ t : t \in r \text{ and } t \notin s \}$

r	Rname	Rid
	A	1
	B	2
	D	3
	F	4
	E	5

s	Rname	Rid
	A	1
	C	2
	D	3
	E	4

$\Rightarrow r - s$

	Rname	Rid
	B	2
	F	4

Cartesian Product

$r \times s = \{ t : t \in r \text{ and } t \in s \}$

r	A	B
	1	2
	3	4

s	C	D
	5	6

A	B	CD
1	2	68
3	4	68

Project

In order to succeed, your desire for success should be greater than your fear of failure

24 Saturday

Wk 04 - Day 024

JANUARY '15

	M	T	W	T	F	S	S	M	T	W	T	F	S	S		
						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											

Additional operation

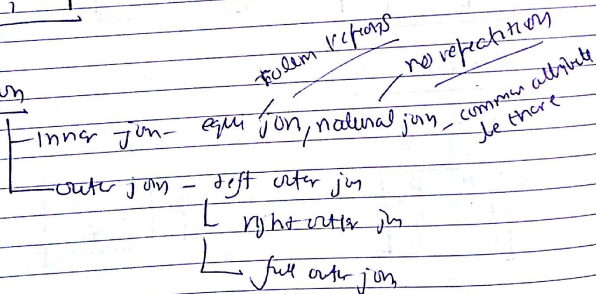
(1) Addition

(i) Intersection

$r \cap s = \{ t: t \in r \text{ and } t \in s \}$

r		s		$r \cap s = r - (r - s)$
A	B	A	B	
1	1	2	1	
1	2	2	4	

(2) Join



25 Sunday

Join is cartesian product followed by selection process

Priorities

Success makes success, like money makes money

JANUARY '15

	M	T	W	T	F	S	S	M	T	W	T	F	S	S
						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									

Monday 26

Wk 05 - Day 026

~~= dangling tuple~~

- dangling tuple - which is absent after natural join

(2) outer join - based on which matched & unmatched data

def OJ - return tuples available in left side table with matched data of 2 tables & null for right table's column.

$r \bowtie s$ or $r \bowtie_{\text{key}} s$

r1	emp	dept	n	dept	real
Smith	sales		hr	mon	
Black	hr		product	Brown	
White	hr				

$r \bowtie r2$

emp	Dept	real
Smith	Sales	NULL
Black	hr	Mon
White	hr	hr

Priorities $r \bowtie_{\text{key}} s$ return tuple in right table

emp	dept	real
Black	hr	mon
White	hr	hr
NULL		hr

If at first you don't succeed, think how many people you've made happy

27 Tuesday

WK 05 - Day 027

Full OJ

r = ~~SQL~~

Emp	Dept	Job
Smith	Sales	Mem
Black	Mr	Man
White	Mr	Man
Null	Prod	Brw

#

SQL

TCL

DDL

DML

DCL

data control lg

- create new table or schemas
- Alter
- Drop

- select
- insert
- update
- delete

- grant
- alter user
- access privilege
- revoke

Priorities

To respond is positive, to react is negative

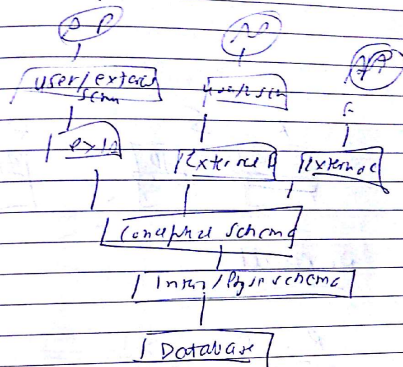
JANUARY '15

M	T	W	T	F	S	S	M	T	W	T	F	S	S
					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						

Wednesday 28

WK 05 - Day 028

- Join query
- Architecture pic
- PL/SQL



Priorities

You are the only one who can use your ability It is an awesome responsibility

29 Thursday

Wk 05 - Day 029

JANUARY '15

M	T	W	T	F	S	S	M	T	W	T	F	S	S
						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					

Relational Algebra

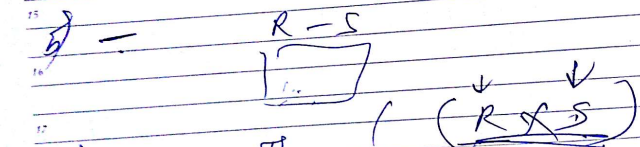
1) σ

2) π

3) \cup



4) \cap No result



5) \times π_{cust} $(R \times S)$

6) $\pi_{\text{custname}} (\sigma_{\text{amount} > 1200} (R \times S))$

7) Division $(\frac{\sigma}{\sigma})$ [all]

Every failure is a step to success

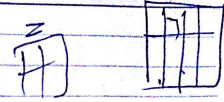
JANUARY '15

M	T	W	T	F	S	S	M	T	W	T	F	S	S
						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								

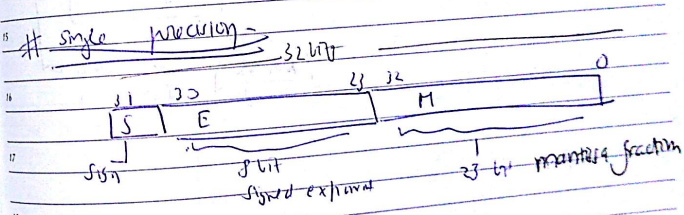
Friday 30

Wk 05 - Day 030

Joins



- 1) Natural
 - 2) Theta
 - 3) Equi
 - 4) Outer
- Inner
- Left
Right
Full



Priorities

"A great social success is a pretty girl who plays her cards as carefully as if she were plain"

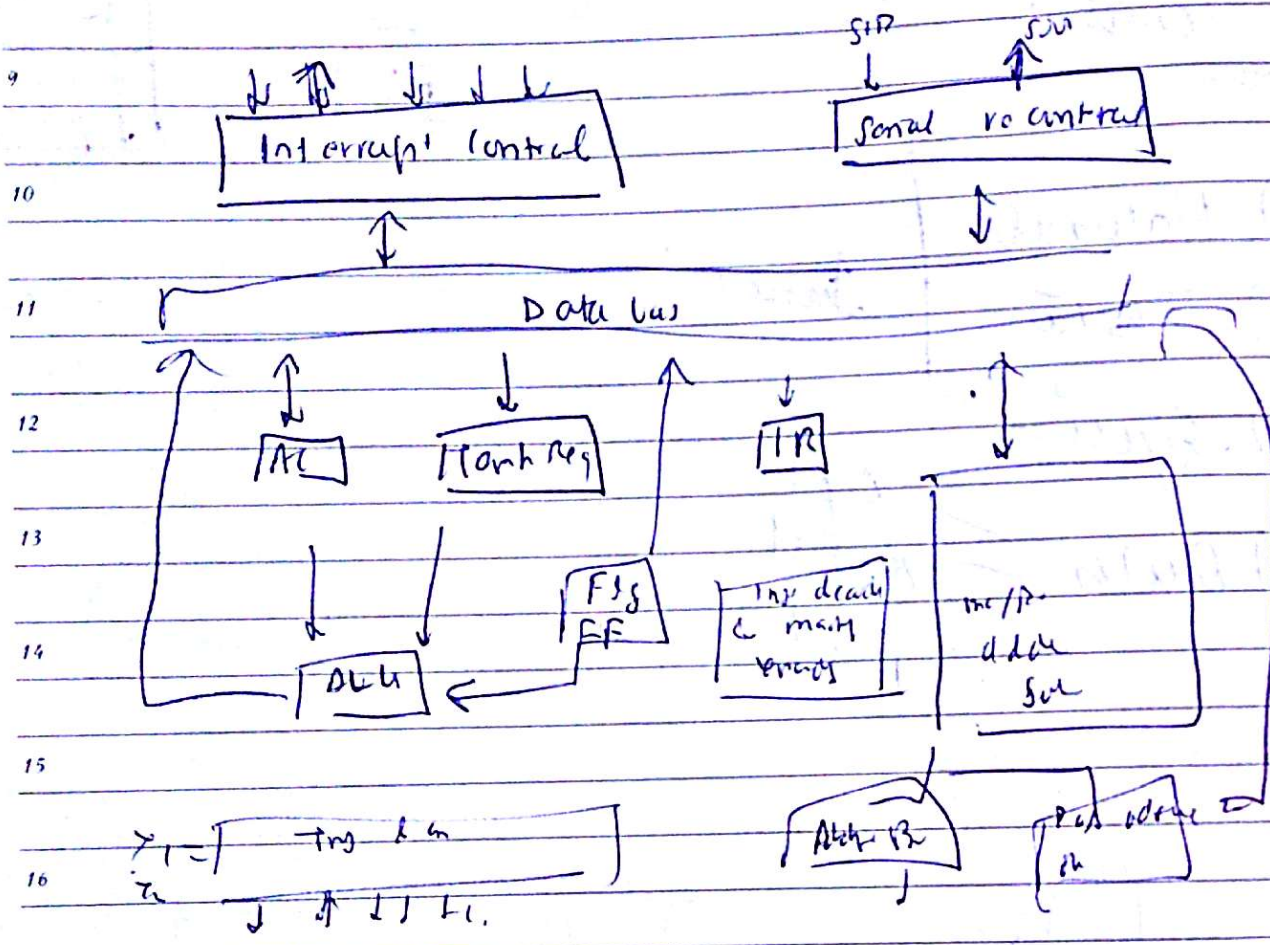
GOOD WRITE

31 Saturday

Wk 05 • Day 031

JANUARY '15

J	M	T	W	T	F	S	S	M	T	W	T	F	S	S
15						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						



Priorities

Successful people are simply those with success habits

Unit-3

Wednesday 04
 06.06 Day 035

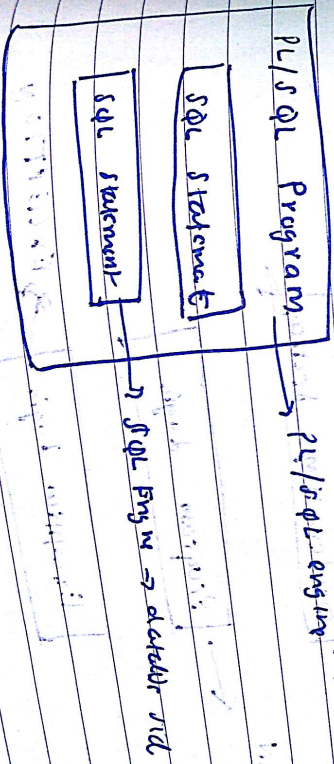
PL/SQL

Procedural lang extension of SQL

SQL

PL/SQL

- Create table
- Manage data
- Control data
- Transaction
- Conditional instruction
- loops
- exception handling
- Store values

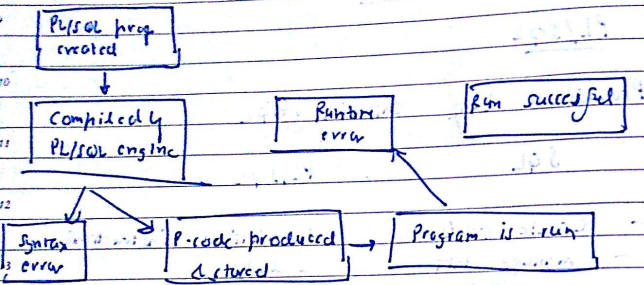


To me success would be to be able to do your very best in everything you do

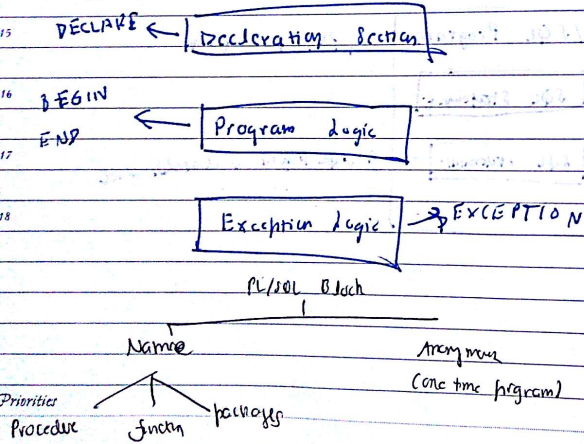
05 Thursday

FEBRUARY
 M T W T F S S M T W T F S S
 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

Week 06 - Day 036



Program Structure



Women will never be as successful as men because they have no wives to advise them

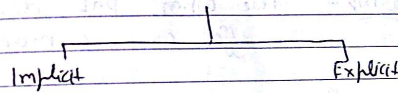
Friday 06

FEBRUARY
 M T W T F S S M T W T F S S
 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

Week 06 - Day 037

Cursor

- Used to process individual rows returned by database system for a query
- Cursor holds rows returned by SQL statement
- Cursor is defined as work area whose SQL statement is executed



- predefined
 - associated with DML
- user defined
 - associated with SELECT statement that return more than one row
- It is called SQL cursor
- Name of implicit cursor is SQL
 - Declared in declaration section & used in execution section
- Attributes are used
 - % FOUND
 - % NOT FOUND
 - % UOPEO
 - % ROW COUNT
- Commands
 - OPEN
 - FETCH
 - CLOSE

Priorities

Success is the progressive realization of a worthy goal or ideal

07 Saturday

Wk 06 • Day 038

FEBRUARY '15

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

Implicit Cursor:-

SQL % FOUND - Boolean attribute return T/F
- True when most recent DML affects one or more rows

SQL % NOT FOUND - True when DML does not affect one or more rows

SQL ROWCOUNT - Return no of rows affected by DML

BEGIN

```

UPDATE Employee
SET Salary = Salary + 100;
IF SQL%NOTFOUND THEN
    dbms_output.put_line ('Employee not');
ELSE IF SQL%FOUND THEN
    dbms_output.put_line ('SELECTED');
END IF;
END;

```

SELECTED

- PL/SQL procedure successfully completed

Priorities

For true success ask yourself these four questions: Why? Why not? Why not me? Why not now?

FEBRUARY '15

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

Monday 09

Wk 07 • Day 040

Explicit

Steps	Syntax	Example
1) Declare the cursor	CURSOR cursor_name IS SELECT	CURSOR CI IS SELECT EName FROM Employee WHERE D_ID=20;
2) Open cursor	OPEN cursor	OPEN CI;
3) Fetch cursor	FETCH (CURSOR) INTO var1, var2	FETCH CI INTO VENAME
4) Close the cursor	CLOSE cursor_name	CLOSE CI

```

as DECLARE
VENAME VARCHAR(20);
CURSOR CI IS SELECT EName
FROM Employee
where D_ID = 20;

```

FD	EName	D-I
11	AA	20
12	BB	10

BEGIN

```

OPEN CI;
LOOP

```

```

FETCH CI INTO VENAME;

```

... statement to exit loop
Ask yourself the secret of YOUR success. Listen to your answer, and practice it
DBMS_output.put_line ('VIEW DML');
END LOOP; END;

OUTPUT AA

Priorities

10 Tuesday

Wk 07 - Day 041

FEBRUARY '15

	M	T	W	T	F	S	S	M	T	W	T	F	S	S
						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									

Stored Procedure -

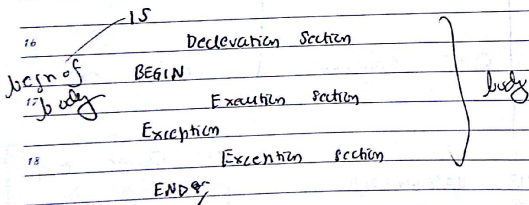
- A named PL/SQL block which performs one or more specific tasks.
- It has header and body.

Header - consist of name of procedure and variables passed to the procedure

Body - The body consist of declaration section, program logic, execution section

Syntax

CREATE [OR REPLACE] PROCEDURE Procname [List of parameter]



eg → CREATE PROCEDURE Emp;

```

IS
  VEMP varchar(20);
  CURSOR C1 IS SELECT E.name
                FROM employee
                WHERE P.ID = 20;
  
```

Don't think of it as failure. Think of it as time-released success

FEBRUARY '15

	M	T	W	T	F	S	S	M	T	W	T	F	S	S
						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						

Wednesday 11

Wk 07 - Day 042

BEGIN

same as before

F.D. is a FUNCTIONAL DEPENDENCY.

- FD. is a constraint that describes relationship between attributes in a relation.

eg R(A, B, C)

F = { A → B, B → C }

F+ = { A → C, A → B, B → C, A → A, B → B, C → C }

Key → X is a key

iff FD all attributes in R
⇒ X should be minimal

Armstrong Axiom and Inference Rules

Armstrong's Axiom = Three distinct axiom of R (X, Y, Z)

Priorities

Success as I see it, is a result, not a goal

12 Thursday

Wk 07 - Day 043

R(X, Y, Z)

FEBRUARY '15

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

Reflexivity	Augmentation	Transitivity
if $A \subseteq B$, then $B \rightarrow A$	if $X \rightarrow Y$	if $X \rightarrow Y, Y \rightarrow Z$
So eg $X \rightarrow Y, Y \rightarrow Z$	then $X \rightarrow YZ$	then $X \rightarrow Z$
$X \rightarrow Y, Y \rightarrow Z$		
$X \rightarrow Z$		

Inference Rules :-

Union	Pseudo Transitivity	Decomposition
if $X \rightarrow Y$ $X \rightarrow Z$ then $X \rightarrow YZ$	if $X \rightarrow Y$ $Y \rightarrow Z$ then $X \rightarrow Z$	if $X \rightarrow YZ$ then $X \rightarrow Y$ $X \rightarrow Z$

Attribute Closure and Key

α - Closure \rightarrow Using Armstrong axioms

Priorities $\rightarrow R(A, B, C, D, E)$

$F = \{A \rightarrow D, D \rightarrow B, B \rightarrow C, E \rightarrow B\}$

A great leader's courage to fulfill his vision comes from passion, not position

FEBRUARY '15

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

Friday 13

Wk 07 - Day 044

A^+ (A closure) \rightarrow It doesn't determine all attributes of relation so it is not a key

$A^+ \rightarrow ABC$

$D^+ \rightarrow DBC$, $B^+ \rightarrow BC$, $F^+ \rightarrow FBC$, $C^+ \rightarrow C$

So, these closure cannot be a key

$AE^+ \rightarrow$ determine all attributes so $AE \rightarrow$ key

also $AECT^+ \rightarrow$ not a key

Functional Dependency for Key

Q1 $R(A, B, C)$
 $F = \{A \rightarrow B, B \rightarrow C\}$

$A^+ \rightarrow$ so A can be key

only L	LTR	only R
A	B	C

\rightarrow may or may not be key

Priorities $R(A, B, C, D)$
 $F = \{A \rightarrow B, C \rightarrow D\}$

L	LTR	R
A	BC	D

$A^+ \rightarrow AB$, $C^+ \rightarrow CD$

$AC \rightarrow$ key

Success is simple. Do what's right, the right way, at the right time