23.02.2019

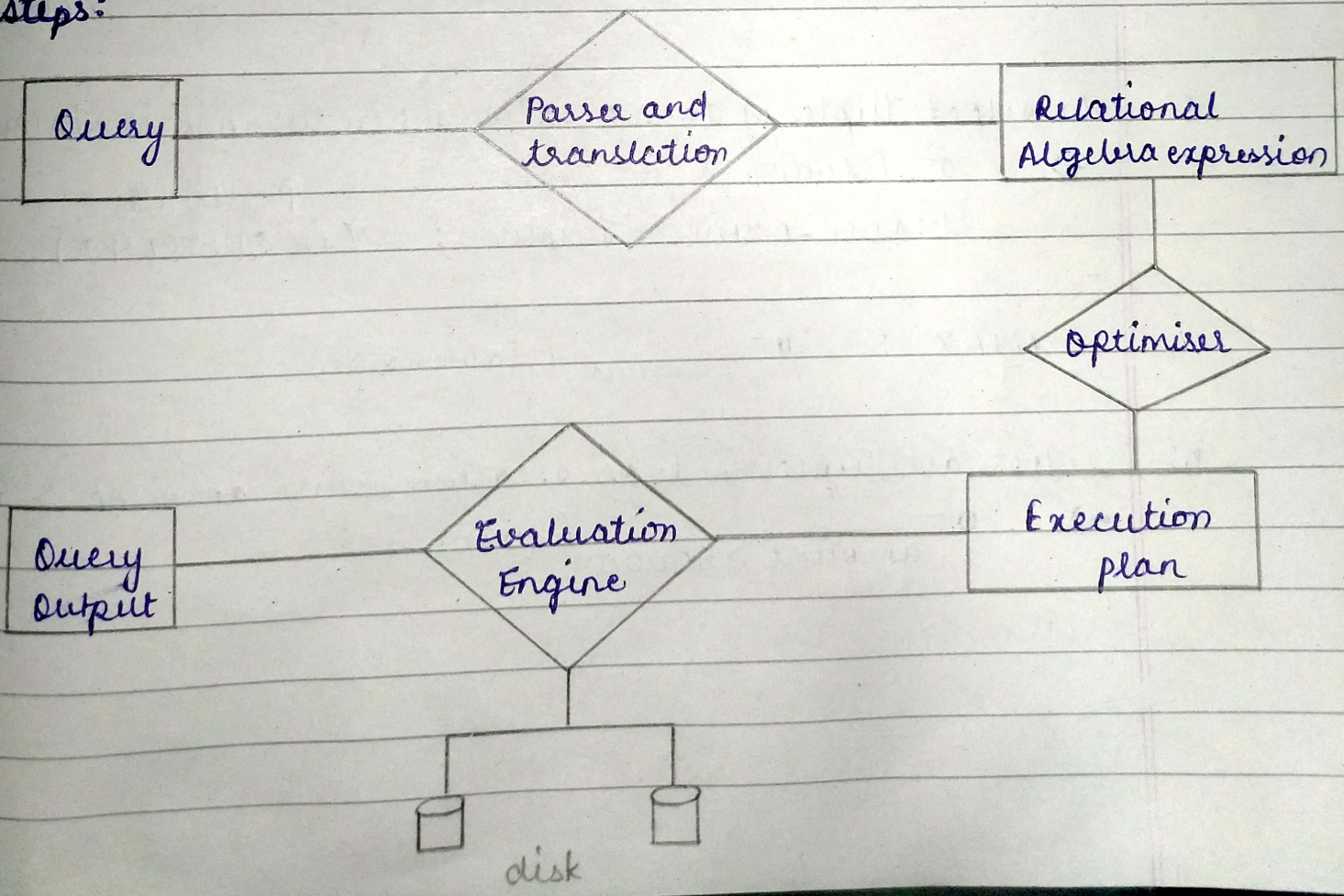
PAGE NO.
DATE:

QUERY PROCESSOR

- Query processing refers to the range of activities involved in extracting data from database.

- **ACTIVITIES :**
  1) Translation (High level language to Machine level language) or Expressions
  2) Optimisation (processing should be efficient and effective)
  3) Evaluation

- **Steps:**



PAGE NO.
DATE:

* **Relation Algebra** is a procedural query language. It collects relations as input and produce results

- **FUNDAMENTAL OPERATIONS OF RA :- (Relation Algebra).**

i) Select ,
ii) Project ,
iii) Union ,
iv) Set difference ,
v) Cartesian Product and
vi) Rename

$\wedge$ → and (Carat)
$\vee$ → or

* **i) Select (σ) :** Sigma

This operation selects tuples that satisfies a given predicates. (conditions).

Q: To select tuples of loan relation where branch is JAIPUR :

-> σ (loan)
branch-name = 'Jaipur' ; (Predicate in SUBSCRIPT)

SYNTAX : $\sigma_{predicate}$ (Tablename)

Q: select all tuples in loan relation where amount > 50000.

-> $\sigma_{amount > 50000}$ (loan)

PAGE NO.
DATE:

* II) **Project (Π) :**
This operation allows to produce a relation.

Q: select LOANNO. and AMOUNT from loan table :
→ $\Pi_{loanno, amount}$ (loan)

Q: find all loanno from loan table where ₹ is > 50000 and branch = Jaipur.
→ Π_{lno} ($\sigma_{amount > 50000 \wedge branch = 'Jaipur'}$ (loan))

* III) **Cartesian Project (⋈) :**
This operation is used to combine/join more than 1 tables.

TABLES : loan (LNO, AMOUNT, BRANCH)
Customer (C_ID, NAME, ADDRESS)
Borrower (C_ID, L_NO)

Q: Find all CUST_ID for customers who are having loan from JAIPUR :
→ Π_{cust_ID} ($\sigma_{loan.lno = borrower.lno \wedge where\ branch = "Jaipur"}$ (loan ⋈ borrower))

IV) **Union :**
Q: Find all loan_no that are in loan and borrower table :
→ Π_{loanno} (loan)
UNION
Π_{loanno} (borrower)

11.03
2019

PAGE NO.
DATE:

JOINS

* Joins are used to work with multiple tables.
  i) Inner Join
  ii) Cross Join
  iii) Outer Join (left, right, full)

*(i) also called EQUI JOIN because the where clause is used to compare 2 columns of 2 tables.

EMP

eno	ename	dno

DEPT

dno	dname

EMP × DEPT

Q: Find empno and their dname from EMP and DEPT tables:

-> select ENO, DNAME
from EMP inner join DEPT
ON emp.dno = dept.dno;

inner_join
(spacing in between

*(iii) Outer joins are used in situations where it is desired to select all rows from the table on the left / right or both.

EMP

eno	ename	dno
1	ABC	D1
2	PQR	D2

DEPT

dno	dname
D1	MK
D2	HR
D3	PD

PAGE NO.
DATE:

EMP × DEPT

eno	ename	dno	dno	dname
1	ABC	D1	D1	MK
1	ABC	D1	D2	HR
1	ABC	D1	D3	PD
2	POR	D2	D1	MK
2	POR	D2	D2	HR
2	POR	D2	D3	PD

all — these values should be kept NULL.

*(ii) Cross Join returns the output of cartesian Product.

EMBEDDED SQL

PAGE NO.
DATE:

* It is an embedding of SQL into procedural language (C) preferred as host language.
referred

1) Exec SQL include SQLCA
2) Exec SQL begin declare section
host_name characterstring (20)
host_empno integer
Exec SQL end declare section
3) Exec SQL whenever SQL error stop
4) Exec SQL connect DB (database name).
5) Exec SQL
select name, empno
into host_name, host_empno
from employee
where eno = 1;

copying name and empno into host_name and host_empno

6) Exec SQL disconnect (used to disconnect with SQL)

PL/SQL

PAGE NO.
DATE:

13.03.2019

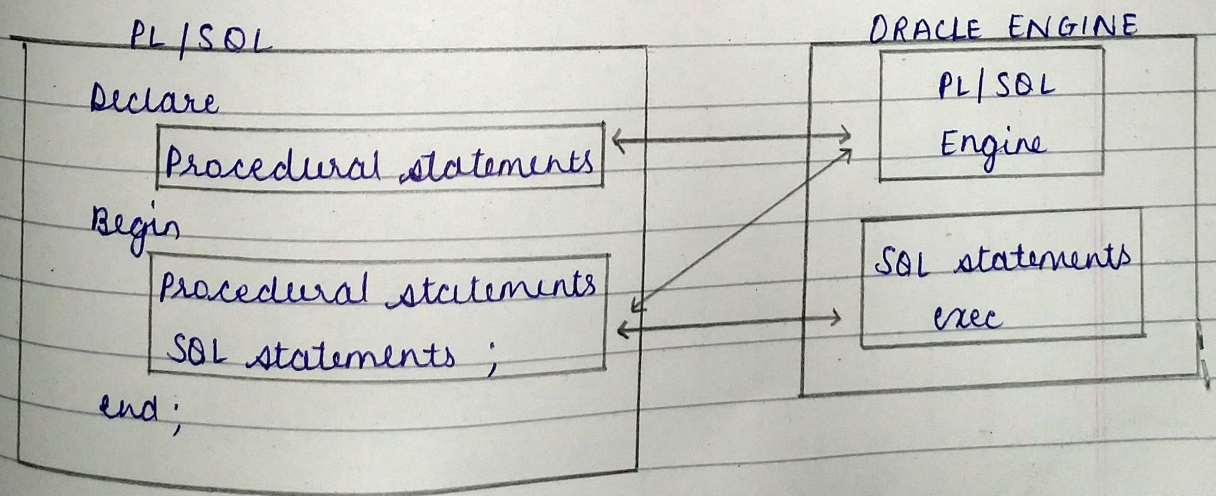
Procedural Language / Structured Query Language.

It is a block structured language that enables a developer to combine power of SQL + Procedural language.
It bridges the gap between database technology and procedural programming language.

* BLOCK STRUCTURE *

```
declare
        declaration of variables ;
begin
        PL/SQL executable statements ;
end ;
```

* EXECUTION ENVIRONMENT *



BCA-II

DBMS

PAGE:
DATE: / /

14.03.2019

NORMALIZATION

UNIT-IV

* Normalization of data can be considered as process of analyzing the given relational schema based on functional dependencies and Primary Key.

To achieve properties like:

- Minimizing redundancy.
- Minimizing the insertion, deletion and Update anomalies

Anomalies

EMPDEPT

Emp_ID	Name	Dept_ID	Dept_Name	
1	ABC	D1	HR	
2	XYZ	D2	MK (and Sales)	(2)
3	PQR	D1	HR	
4	ADD	D2	MK (and Sales)	(4)
NULL	NULL	D3	PD	(5)

- (1) Insertion Anomaly:
  the 5th row is considered as Insertion Anomaly.
- (2) Update Anomaly:
  where updation is to be done multiple times
  the 2nd and 4th rows are considered as Update Anomaly.
- (3) Deletion Anomaly:
  Data loss occurs.

PAGE NO.

PAGE :
DATE : / /

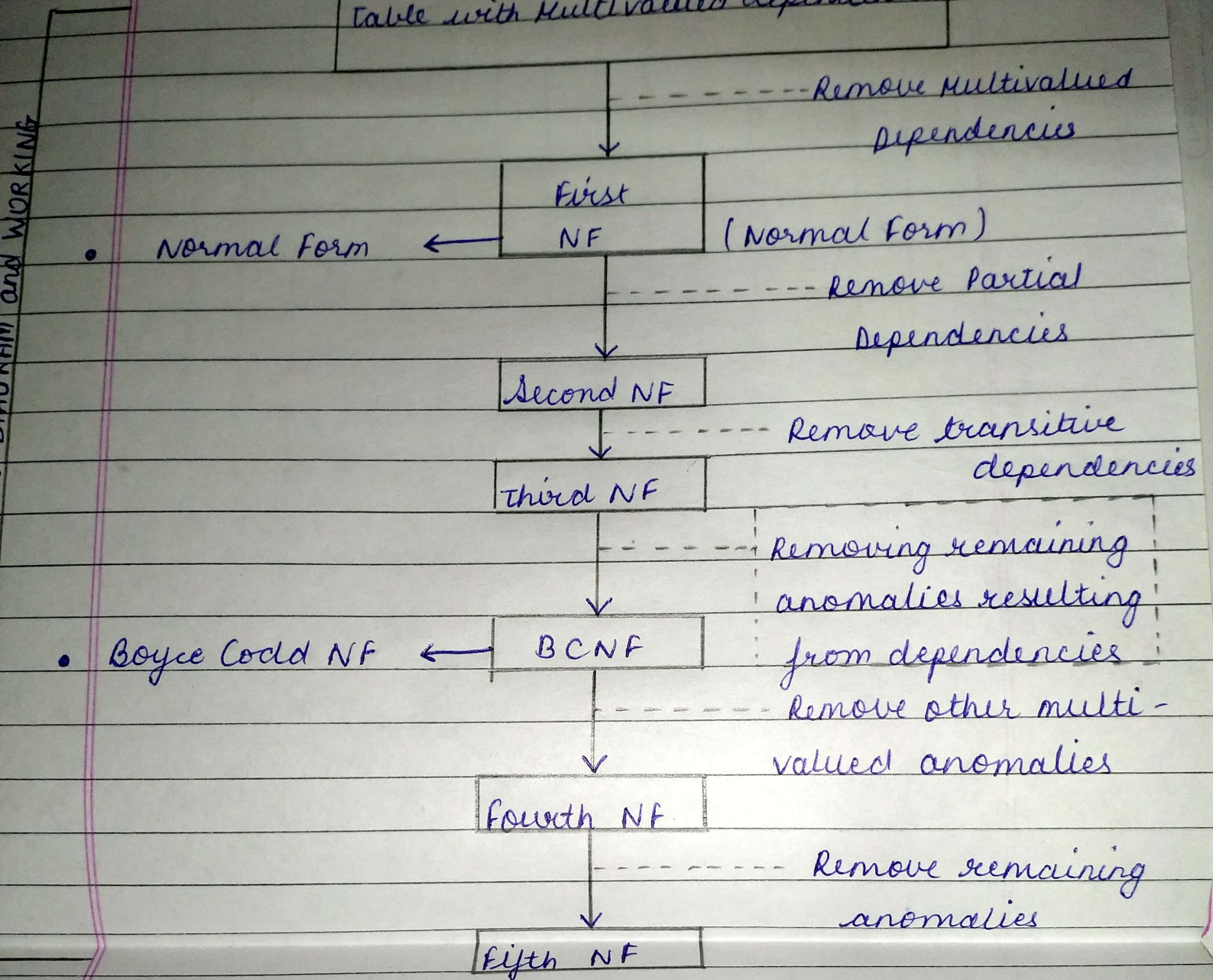


Table with Multivalued dependencies
Remove Multivalued Dependencies
First NF
(Normal Form)
Normal Form
Remove Partial Dependencies
Second NF
Remove transitive dependencies
Third NF
Removing remaining anomalies resulting from dependencies
BCNF
Boyce Codd NF
Remove other multi-valued anomalies
Fourth NF
Remove remaining anomalies
Fifth NF