DTS-22319

Unit III: Interactive SQL and Advance SQL SQL Performance Tuning.

(Weightage - 14 marks)

Oracle Built in functions. (oracle)

- Single Row Functions : Single row or Scalar functions return a value for every row that processed in query.
- Group Functions : These functions group the rows of data based on the values returned by query, example, total average.

3.1 : Inbuilt functions : String, Arithematic.

String Functions

Questions
- Explain any four string functions with example. (5-23) (Um)
- Explain any four string functions. (Um)
- List any four string functions. (2m)
- String functions (Initcaps trim) (2m)

⇒ Functions

- Lower – All letters in upper/lower case
- Upper – Upper case
- Initcap – inital/first letter captial
- LTRIM – Trim from Left
- RTRIM – Trim from Right
- TRIM – To reduce to string.
- SUBSTR – Return 'n' numbers of charachter from position
- LENGTH – Num. of charachters is returnd
- LPAD – Left pad
- RPAD – Right pad.

IPLUTS

Table Name - Person

Last Name	First Name	Address	City
Sharma	Oli	GM Road 10	Pune
Mante	Edison	Camp 23	Pune
Johnson	Kari	RTO Road 20	Banglore

① Initcap : It Display initial letter capitcal (First)

Syntax : Initcap(string)

Example : Select initcap(City) from Person;

Output ⇒

City
Pune
Pune
Banglore

② Lower and Upper :

It Display the string into either lower or upper case.

Syntax : lower (string)
upper (string)

Example : Select lower(City) from Person
Select upper(City) from Person

Output ⇒ Lower

city
pune
pune
Banglore

upper

City
PUNE
PUNE
BANGLORE

③ TRIM

Trim (trim_text from value).

Syntax : - trim(string, trimtext)

Example : Select trim('o') from person

⇒ All 'o' from table will be deleted.

trim and Rtrim

It trims or cuts the charachter from left or right.

Syntax : ltrim(string, trimtext)
rtrim(string, trimtext)

Example : Select ltrim('Pune', 'P') from Person;
Select Rtrim('Pune', 'e') from Person;

Output

City
Une
Une
Banglore

City
Pun
Pun
Banglore

⑤ Substr(Main

It returns the substring from specified position

Syntax : Substr(Main string, Position, Charachters to be replaced) (-)

Example : Select Substr(City, 1, 3) from Person;

⇒

City
Pun
Pun
Ban

⑥ Lpad and Rpad

It is used for formatting from left and right side by using any charachter. example: mob no of 10 digit

Syntax

lpad(main string, length, charachter to be padded)
Rpad(main string, length, charachter to be padded)

Length = size we want

Example Select lpad(City, 15, '*') from person;

or

City	City
~~**********~~	Pune~~*****~~

⑦ Concatenation (merging two columns)

It is used to concatenate several expression together.

Syntax: Concat(expr1, exp2, expr3, ...)

Example: select concat(firstName, City) from person;

Oli Pune
Edison Pune
Kari Banglore

⑧ Length: It is used to calculate the length of given string.

Syntax: Length(String)

Example: SELECT length(City) from Person.

City
4
4
8

Arithematic

They are also known as Numeric functions, They are used to perform operations on numbers.

* power(4,2) 4
* SQRT(25) √25
* mod(

① Abs

This SQL ABS() returns absolute value of number passed as argument.

Exp: Select ABS(-17.38) from dual;

-ve ⇒ +ve

⇒ 17.38

② CEIL

This SQL CEIL() will round up any +ve or -ve decimal within higher.

-17.38

⇒ 18

③ Floor() = nearest no round up

17.38 ⇒ 17
17.56 ⇒ 18

④ Trunc(x,y)

(125.456, 1) 2

⇒ 125.4 125.45

⑤ Round(x,y)

(140.235, 2)

140.25

⑥ Exp(x) ⇒ Exponential form

2 : Date and time, Aggregate functions.

Date and time

These are functions that takes value that are of datatype DATE as input and return value of data type DATE. except for the months_between functions, which return a number as output.

① Add_Months :- Adds the specified numbers of months to the date value.

Syntax :- Add_months(date, numbers of months)

Example :- select add_months(sysdate, 5) from dual;

This will add 5 months to existing date.

② Greatest :- Returns the greatest value in list of expression.

Syntax :- ~~Greatest~~(expr1, exp2, expr3)

Example :- Select Greatest ('10-JAN-07', '12-OCT-07') from dual;

output → 12-Oct-07

③ Least :- Returns the least value in list of expression.

Syntax - Least (expr1, expr2, expr3)

Example - Select least ('10-Jan-07', '12-Oct-07')

Output - (10 Jan -07)

④ Datediff :

Returns the difference in days between two dates value.

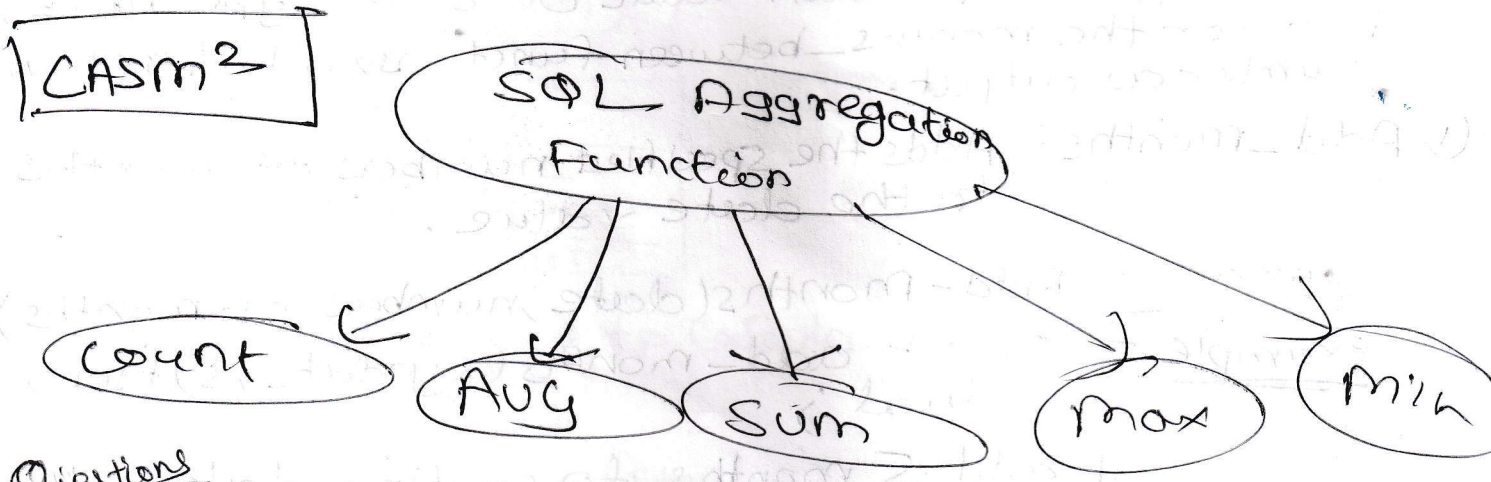
Syntax : Date diff (date1, date2)

Example, Select datediff ('10-JAN-07', '12-OCT-07 from dual

O/P = 276

Aggregate Functions

* SQL → aggregation function is used to perform the calculations on multiple rows of a single column of table. It returns a single value.
* It is also used to summarize the data.



Questions

- Explain any four aggregate function with example - 4marks
- State the use of AVG function with example.(2
- Enlist four aggregate functions (2M)

⇒ Table Name Person

LastName	FirstName	Age	City
Sharma	Oli	34	Pune
Mante	Edison	14	Pune
Johnson	Kari	19	Banglore

① **Count** : The count function returns the numbers of row without a null value in specified column

Syntax: Select count(column) from tablename;

Example : Find the no. of persons with a value in the age filed in the person table..

⇒ Select count(Age) from Person;

* Count → function → returns no. of selected rows in selection.
  - → Select count(*) from table_name;
  - → Select count(*) from person; → 3
    Select count(* from person where age > 20 → 2

Average(AVG): The AVG function returns the average value of a column in a selection NULL values are not included in the calculation.

$\left(\frac{A+B}{2}\right)$ - Avg ← Example

Syntax: Select avg(column) from tablename;

Example: Find the average of persons in the person table

⇒ Select avg ~~(City)~~ (Age) from persons;

↳ O/P ⇒ 32.67

~~Example~~

Find the average of persons that are older than 20yrs.

⇒ Select avg(Age) from persons where Age > 20;

↳ O/P = 39.5

(3) Sum: The Sum function returns the total sum of a column in a given selection. Null values are not included in calculation.

(+)

Syntax: Select Sum (Age) from persons ⇒ 98

↳ Select Sum (column) from table name;.

Select Sum(Age) from persons where age > 20 ⇒ 20

(4) Max Function: The max function returns the highest value in a column. Null values are not included in Calculation.

Syntax: Select max(column) from table_name;

Example: Find the maximum age from person table.

Select max(age) from persons;

Value
45

(5) Min Function: The min function returns the lowest value are not included in calculation.

Syntax: Select min(column) from table-name;

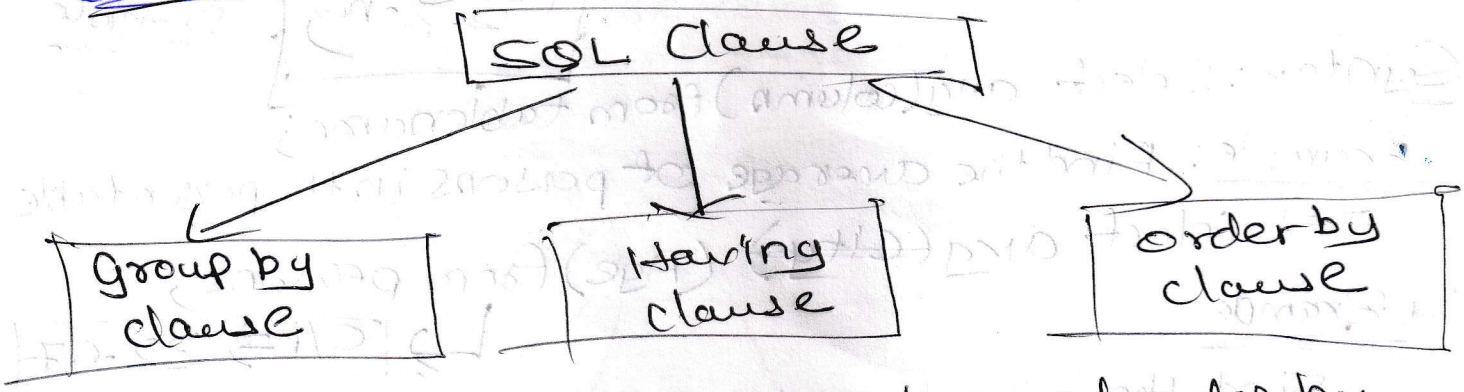
Find the minimum age from person table

⇒ Select min(Age) from persons;

Value
19

3.3 Queries using Group by, having and order by C
Joins - Inner and Outer Join, Subqueries

Clauses



Question) State the use of group by and order by clause — (4) marks

⇒ **Group by**

- SQL GROUP BY statement is used to arrange identical data into groups. The GROUP BY statement is used with the SQL SELECT statement.
- The GROUP BY statement follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.
- The GROUP BY statement is used with aggregate function

~~Example~~ **Syntax**: select column, sum(column) from table group by column;

Example

table - Sales

Company	Amount
Wipro	5500
IBM	4500
Wipro	7100

Fig 1

⟹

Company	Amount
Wipro	17100
IBM	17100
Wipro	17100

SQL >> Select company, sum(Amount) from Sales

SQL >> Select company, sum (Amount) from Sales GROUP BY company;

Company	Amount
Wipro	12600
IBM	4500

Having Clause

Having clause was added to SQL because the where keyword could not be used against aggregate functions (like sum) and without having clause would be impossible to test for result conditions.

Syntax:- Select column, sum(column) from table group by column Having ~~sum~~ sum(column) condition value;

Fig 1 = reference

Select company, sum(amount) from sales group by company having sum(amount) > 10000;

Company	Amount
Wipro	12600

Order by Clause

* The order by clause sorts the result-sets either in ascending or descending order
* It sorts the records in ascending order by default
* DESC keyword for descending order.

Syntax: Select (column names) from table_name where Condition Order By column name ASC|DESC;

Company	Amount
Sega	3412
MBT	5678
Wipro	2312
Wipro	6798

⟹

Company	Amount
MBT	5678
Sega	3412
Wipro	6798
Wipro	2312

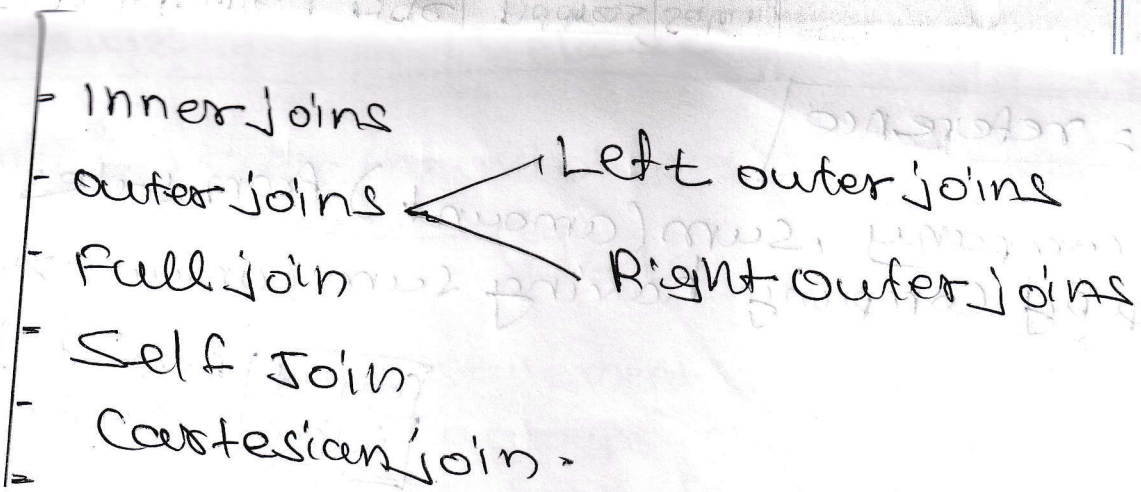
Select Company, Amount From orders Order by company

Reverse : DESC test

Sql Joins

Imp

* As the name shows, JOINS means to combine someth
* In case of SQL, JOIN means "to combine two or more table".
* The SQL JOIN clause take records from two or



INNER JOINS / EQUIJOIN

* The most important and freaquently used of the join is INNER JOINS.
* The INNER JOIN creates a new result by combining column values of two tables (table 1 and table 2) based upon the join-predicate.
* The query compares each row of table 1 with each row of table 2 to find all pairs of rows which satisfy the join-predicate.
* when the join-predicate is satisfied, column values of each matched pair of rows A and B are combined into a result row.

Syntax: Select column name from table 1 named
Inner join table 2 name on table1.common field =
table2.common field

Example: Select I D, Name, Amt, Date from
Customers Inner Joint Orders ON
Customers.ID = Orders.customers_ID;

Sql Joins

* As the name shows, JOINS means to combine somet
* In case of SQL, JOIN means "to combine two or more table".
* The SQL JOIN clause take records from two or more tables in a database and combines in together.

Customers - table

ID	Name	Age	Address	Salary
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	Kaushik	23	Kota	2000.00
4	Chaithali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	M.P	4500.00
7	Muffy	24	Indore	10000.00

order - table

OID	Date	Custom-ID	Amt
102	08-10-08 00:00:00	3	3000
100	8-10-20	3	1500
101	9-11-20	2	1560
103	8-05-20	4	2060

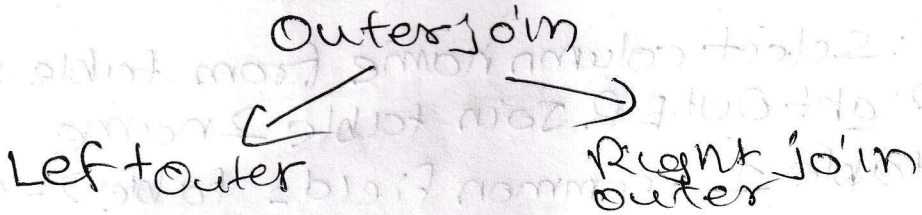
which satisfy the join-predicate.

* when the join-predicate is satisfied, column values of each matched pair of rows A and B are combined into a result row.

Syntax: Select column name from table1 name1 Inner join table2 name on table1.common field = table2.common field

Example: Select I.D, Name, Amt, Date from Customers Inner Joint Orders ON Customers.ID = Orders.customers_ID;

ID	Name	Amount	Date
3	kaushik	3000	09-10-08
3	kaushik	1500	09-10-08
2	khilan	1560	09-11-20
4	chaitali	2060	08-05-20



Left Outer join

* The SQL Left join returns all rows from the left table even if there are no matches in the right table.
* This means that a left join returns from the left table, plus matched values from the right table or NULL in case of no matching join predicate.

Syntax : Select column name from table1 name left outer join table2 name on table1.commonfield = table2.commonfield

Example,

Select ID, Name, Amount, Date from Customers left outer join orders on customer.ID = orders.customer.ID

ID	Name	Amount	Date
1	Ramesh	NULL	NULL
2	khilan	1560	09-11-20
3	kaushik	3000	09-10-08
3	kaushik	1500	9-10-08
4	chaitali kaushik	2060	8-05-20
5	Hardik chaitali	NULL	NULL
6	komal	NULL	NULL

Right outer join

The SQL Right Join returns all rows from the right table, even if there are no matches in the left table. This means that a right join returns all the values from the right table, plus matched values from left table or NULL in case of no matching join predicate.

Syntax: Select column name from table 1 name
Right OUTER Join table 2 name on
table 1 . common field = table 2 . common field.

Example: Select ID, Name, Amount, date from Customers
Right Join orders on customer.ID =
order . Customer ID

ID	Name	Amount	Date
3	kaushik	3000	09-10-08
3	kaushik	1500	08-10-08
2	Khilan	1560	09-11-20
4	chahal	2060	08-05-20

Full join

The SQL Full join combines the result of both left and right outer joins.
The join table will contain all records from both the tables and fill in NULL's for missing matches on either side.

Syntax: ~~Select~~ table 1 name. Full join t2 name
Select column name from
On table 1 . common field = table 2 . common field

ALL records will be inserting + Same records 2 times as much time

Views : Concept of view, Create & Alter, update view

Questions ⟹ Q// Write syntax for creating and dropping views. (4m)

Q// Describe the concept of view with example. State it purpose (4m)

Q// Describe view and write a command to create view. (4m)

Q// Write and explain syntax for creating view with example. (4m)

* Views in SQL are considered as a virtual table. A view also contains rows and columns.
* To create view, we can select the field from one or more tables present in the database.
* A view can either have specific rows based on certain condition or all the rows of table.

Purpose of view

* Structure data in a way that users or classes of users find natural or intuitive.
* Summarize data from tables which can be used to generate reports.
* Restrict access to the data in such a way that a user can see and (sometimes) modify exactly what they need and no more.
* It works on logical level of DBMS.

Table - Customers

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedbad	2000.00
2	Khilan	25	Delhi	1500.00
3	Kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

Virtual Base table

Creating View

* Database views are created using the create v statement.
* View can be created from a single table, multiple table or another view.
* To create a view, a user must have the appropriate system privilege according to the specific implementation.

Syntax: create view view-name as select column from table_name WHERE (Condition)

SQL > create view CSM AS select name, age from customers;

SQL > SELECT ~~view *~~ from ~~cus~~ CSM;

name	age
Ramesh	32
khilan	25
kaushik	23
chaitali	25
komal	27
Hardik	22
Muffy	24

View

Updating → SQL > update view CSM SET Age = 35 where name = 'Ramesh'.

Delete → delete from CSM where age = 22;

Dropping View

* Obviously, where you have a view, you need a way to drop the view if ~~there~~ is no longer needed
* The syntax is very simple → Drop view view-name

example, Drop view CSM;

The whole CSM view table we have created will be deleted.

Alter sequence

* Use the Alter sequence statement to change the increment, minimum and maximum values, cached numbers, and behaviour of existing sequence.

Alter sequence sequence name (// same command as create;)

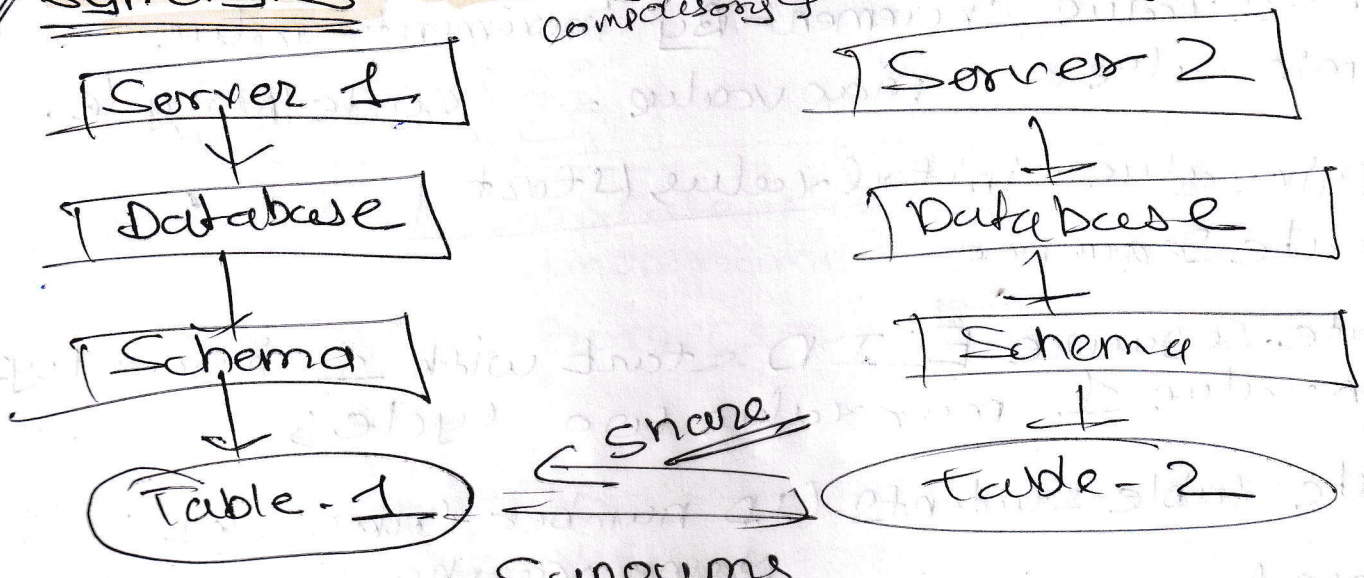
Dropping sequence

If a sequence is no longer / you can drop the sequence using Drop sequence statement.

Syntax: Drop sequence sequence_name;

SQL> Drop sequence E_ID

Synonyms : (creating Synonyms & droping Synonyms.



Synoyms

* A Synoym is an alias means short name for particular time for table, view, snapshot, sequence, procedure, function or package.
* It provide security in terms of Database Administrator
* Synonyms can provide a level of security by masking the name and owner of an object and by providing location transperency for remote object of distributed database.
* They are convient to use and reduce the complexity of sql statement for database users.

* Synonyms <— public / private
* public synonyms: owned by a special user group needs named public is accessible to every user in a database.
* public synonyms: A private synonym contained schema of a specific user and available only to user and user's grantes.

Create Synonyms

* To create synonyms (private) you must have the create synonym privilage.
* To create synonym you must have to create public synonym system privilage.

The Syntax for Creating Synonyms

create synonyms synonym_name for table_name;

Example

=> Create synonym ems for employee;

Synonym created.

=> select * from ems; //to display

You can insert the rows in the synonyms ems as you insert rows in employee table.

Dropping Synonyms

* If Synonym is no longer required, you can drop the sequence using the drop synonym statement.
* The original table for which synonym is created does not get deleted when you deleted for synonyms
* When you drop synonym, its definition is removed from data dictionary.

Syntax: Drop sequence sequence_name;

~ Drop sequence ems;

Indexes : Index types. Creating of an Index, Simple Unique and Composite Index. Dropping Indexes.

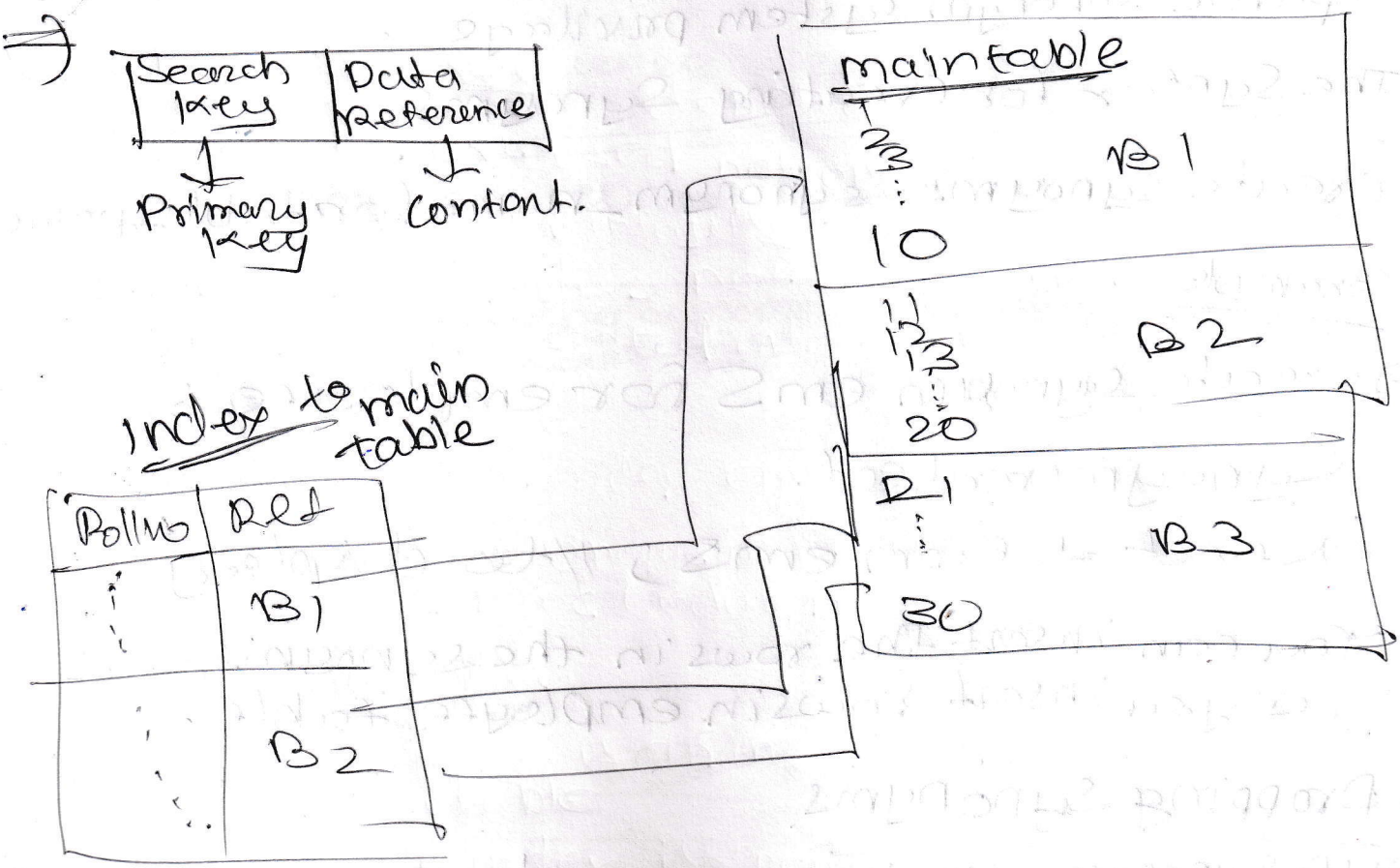
Questions

Q// write and Explain the syntax for creating and dropping indexes with example. (4m)

Q// write syntax for
i) Create Index
ii) Drop Index
} (4m)

Q// Create simple and composite index. write command to drop above index (6m)

⇒



Indexes

* Indexes are optional structure associated with table and groups of database that allows SQL statements to execute quickly against table.
* Just as the index in manual helps you locate information faster than if there were no index, an article,
* An oracle index provides a faster access path to table data.

* You can use indexes without rewriting any queries.
* Being independent structures, they requires storage space.
* You can use indexes without rewriting any queries.
* Being independent structures, they requires storage space.
* You can create or drop an index without affecting the base tables, applications or other indexes.
* Oracle automatically maintains indexes when you insert, update and delete rows of the associated tables.

Create Index

The basic syntax of Create Index is as follows:

```
Create Index index_name ON table_name;
```

Drop Index

An index can be dropped using SQL Drop command. Care should be taken when dropping an index because the performance may either slow down or improve.

Syntax:

```
Drop Index index_name on
table_name;
```

Types of Indexes

- Single Column Indexes
- Unique Indexes
- Composite Indexes

① Single-column Indexes

A single column index is created based on only one table column.
The basic syntax is as follows.

Syntax

Create Index index_name on table name (column name);

Example: Create Index ems on employee (R_no);

② Unique Indexes

do not duplicate any ID

* Unique Indexes are used not only for performance but also for data integrity.
* A unique index does not allow any duplicate values to be inserted into table
* **Syntax**

Create unique index emp on employee(emp_id)

Syntax Create unique index index name on table name (column name);

③ Composite Index

* A composite index is an index on two or more column of a table
* A composite index is known to be concatenated index.

Syntax: Create index index name on table_name (column1, column2);

Create index emp on employee (Emp_id, Dept_ID)