ISO 9001 : 2000 Certified

Society for Computer Technology & Research's

PUNE INSTITUTE OF COMPUTER TECHNOLOGY

LabManual

MarkingScheme

25marks-PracticalExam

25 marks-Term

Work

INDEX

6	**Cursors: (All types: Implicit, Explicit, Cursor FOR Loop, Parameterized Cursor)** Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N_EmpId with the data available in the table O_EmpId. If the data in the first table already exist in the second table then that data should be skipped.			
7	**DatabaseTrigger(AllTypes:RowlevelandStatement leveltriggers,BeforeandAfterTriggers).** Writeadatabase trigger on **Library** table. The System should keeptrack of the records that are being updated or deleted. Theold value of updated or deleted records should be added in**Library_Audit**table. **FrameproblemstatementforwritingDatabaseTriggersof all types, inline with above statement. The problemstatementshouldclearlystatetherequirements.**			
8	**DatabaseConnectivity:** WriteaprogramtoimplementMySQL/Oracledatabaseconnectivitywithanyfrontendlanguagetoimplement Databasenavigationoperations(add,delete,editetc.)			
	GroupB:NoSQLDatabases			
9	DesignandDevelopMongoDBQueriesusing**CRUDoperations**. (Use CRUD operations, SAVE method, logicaloperators)			
10	**MongoDB AggregationandIndexing:** DesignandDevelopMongoDBQueriesusingaggregationandindexingwithsuitableexampleusingMongoDB			
11	**MongoDB Map-reducesoperations:** ImplementMapreducesoperationwithsuitableexampleusingMongoDB.			
12	**DatabaseConnectivity:** WriteaprogramtoimplementMongoDBdatabaseconnectivity withanyfrontendlanguagetoimplementDatabase navigationoperations(add,delete,editetc.)			
	GroupCMiniProject:DatabaseProjectLifeCycle			
13	Using the **database concepts covered in Group A and Group B**, develop an application withfollowingdetails: 1. FollowthesameproblemstatementdecidedinAssignment-1ofGroupA. 2. FollowtheSoftwareDevelopmentLifecycleandother conceptslearntin**SoftwareEngineeringCourse**throughouttheimplementation. 3. Developapplicationconsidering: • FrontEnd:Java/Perl/PHP/Python/ Ruby/.net/anyotherlanguage Backend:MongoDB/MySQL/Oracle.			

AssignmentNo.	**1**
Title	ERModelingandNormalization
PROBLEM STATEMENT/DEFINITION	Decide a case study related to real time application in group of 2-3 students and formulate aproblem statement for application to be developed. Propose a Conceptual Design using ERfeatures using tools like ERD plus, ER Win etc. (Identifying entities, relationships betweenentities, attributes, keys, cardinalities, generalization, specialization etc.) Convert the ERdiagramintorelationaltablesandnormalizeRelationaldatamodel.
Objectives	a) Data Modeling b)converting ERD to table c) Explore ant ERD Tools
Software packages and hardware apparatus used	MySQL/Oracle PC with the configuration as Latest Version of 64 bit Operating Systems, Open Source Fedora-GHz. 8 G.B. RAM, 500 G.B. HDD, 15"Color Monitor, Keyboard, Mouse
References	**DatabaseManagementSystemLaboratory**
STEPS	Refer to details
Instructions for writing journal	• Date • Title • Problem Definition • Learning Objective • Learning Outcome • Theory-Related concept,Architecture,Syntax etc • Class Diagram/ER diagram • Test cases • Program Listing • Output • Conclusion

Subject Co-ordinator
(Mrs. Pranjali Joshi)

Head of Department
(Computer Engg)

AssignmentNo.1

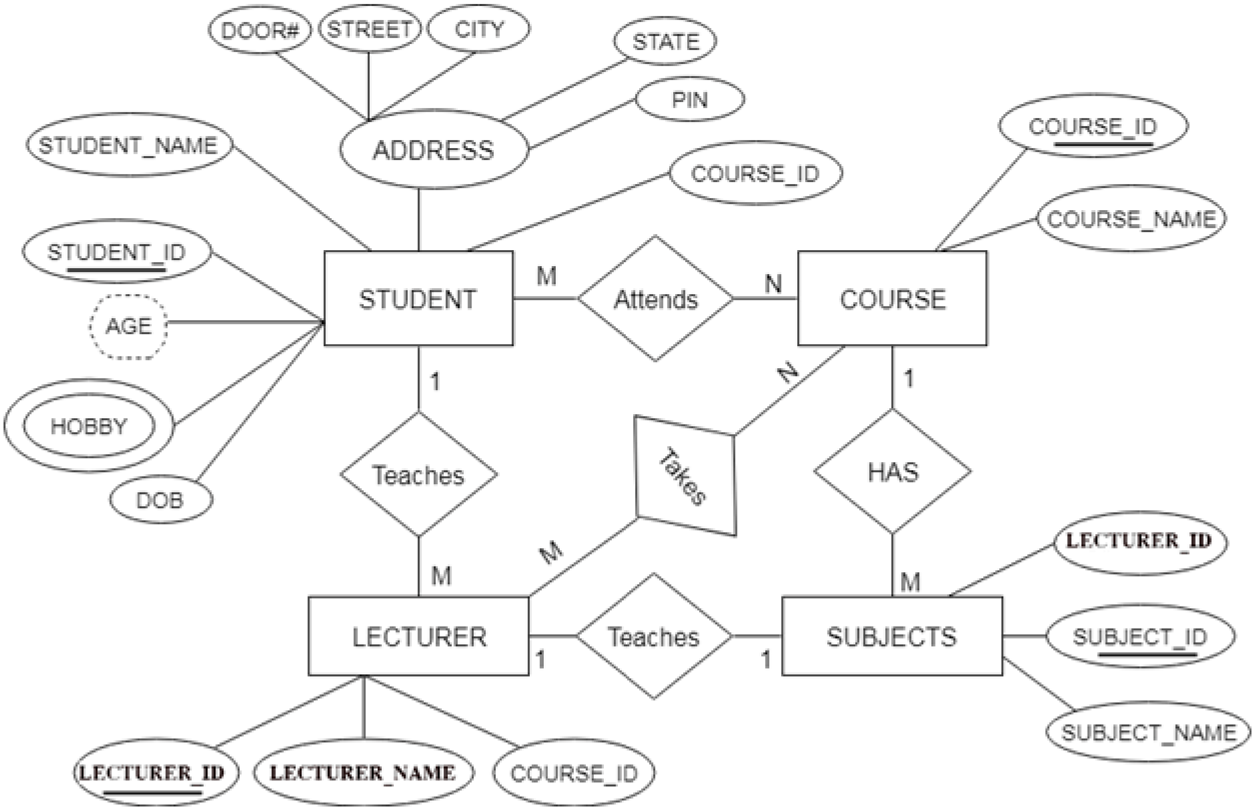
Title : ERModelingandNormalization:

Objectives:a) Data Modeling b)converting ERD to table c) ERD Tools

Theory: Reduction of ER diagram to tables

IntroductionThe database can be represented using the notations, and these notations can be reduced to a collection of tables. In the database, every entity set or relationship set can be represented in tabular form.

The ER diagram is given below:



There are some points for converting the ER diagram to the table:

- **Entity type becomes a table.**

In the given ER diagram, LECTURE, STUDENT, SUBJECT and COURSE forms individual tables.

- **All single-valued attribute becomes a column for the table.**

In the STUDENT entity, STUDENT_NAME and STUDENT_ID form the column of STUDENT table. Similarly, COURSE_NAME and COURSE_ID form the column of COURSE table and so on.

A key attribute of the entity type represented by the primary key.

In the given ER diagram, COURSE_ID, STUDENT_ID, SUBJECT_ID, and LECTURE_ID are the key attribute of the entity.

- **The multivalued attribute is represented by a separate table.**

In the student table, a hobby is a multivalued attribute. So it is not possible to represent multiple values in a single column of STUDENT table. Hence we create a table STUD_HOBBY with column name STUDENT_ID and HOBBY. Using both the column, we create a composite key.

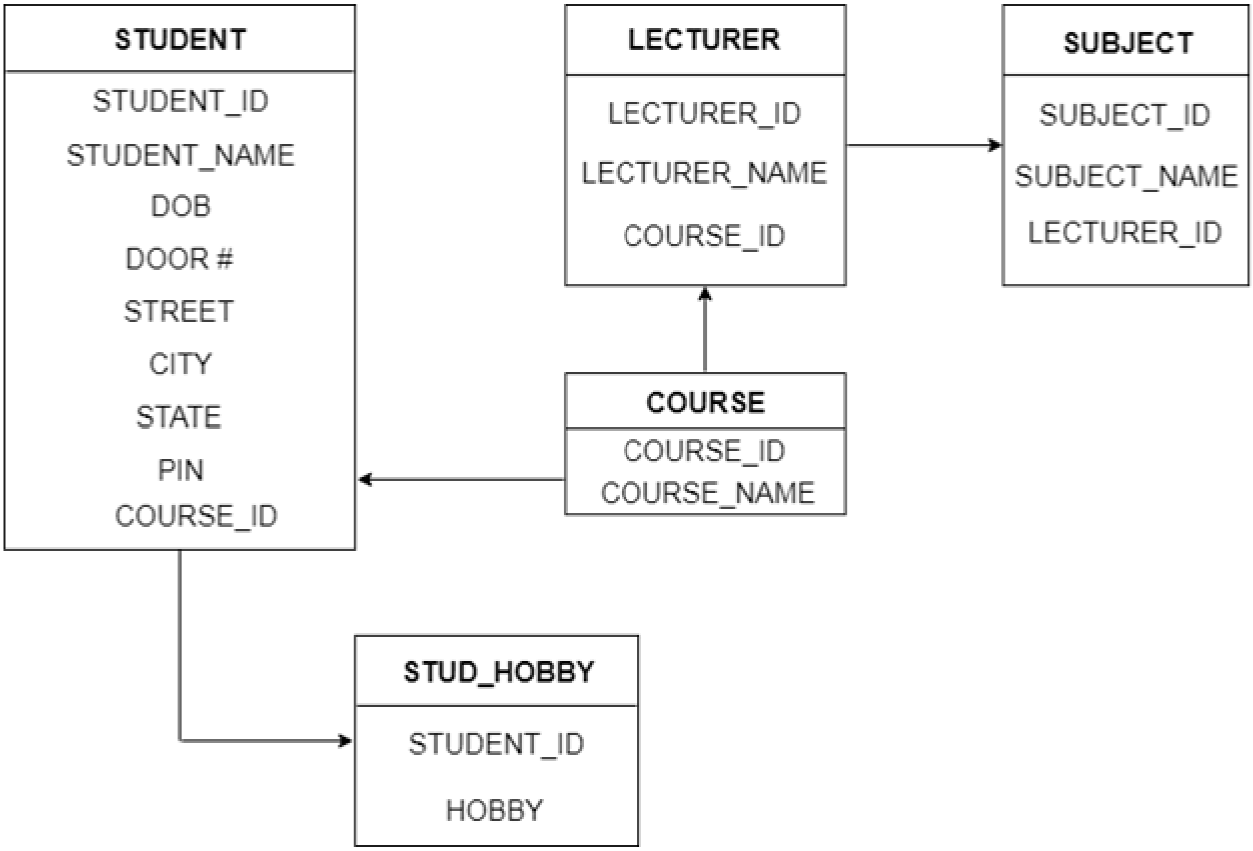
- **Composite attribute represented by components**

In the given ER diagram, student address is a composite attribute. It contains CITY, PIN, DOOR#, STREET, and STATE. In the STUDENT table, these attributes can merge as an individual column.

- **Derived attributes are not considered in the table.**

In the STUDENT table, Age is the derived attribute. It can be calculated at any point of time by calculating the difference between current date and Date of Birth.

Using these rules, you can convert the ER diagram to tables and columns and assign the mapping between the tables. Table structure for the given ER diagram is as below:



RDBMS Terminology:

Before we proceed to explaindatabase system, let's revise few definitions related todatabase.

Database: Adatabase isacollection of tables, with related data.

Table: A table is a matrix with data. A table in a database looks like a simple spreadsheet.**Column**:Onecolumn (dataelement)containsdataofoneandthesamekind,forexampletheColumnpostcode.

Row:Arow(tuple,entryorrecord)is agroupof related data,forexamplethe dataofonesubscription.

Redundancy:Storingdatatwice,redundantlytomakethesystemfaster.

PrimaryKey:Aprimarykeyisunique.Akeyvaluecannot occur twice inone table.With akey,youcan findatmostonerow.

ForeignKey:Aforeignkeyisthelinkingpinbetweentwo tables.

CompoundKey:Acompoundkey(composite key)is akeythatconsists of multiplecolumns,Becauseonecolumn isnotsufficientlyunique.

Index:Anindexina databaseresemblesan indexattheback ofabook.

ReferentialIntegrity:ReferentialIntegritymakessurethataforeignkeyvaluealwayspointstoAn existingrow.

isafast,easy-to-useRDBMSbeingusedformanysmallandbigbusinesses.

isdeveloped, marketed, and supported by My SQLAB, which is a Swedish company.isbecomingso popularbecauseof manygoodreasons:

- isreleasedunderanopen-sourcelicense.Soyouhavenothingtopaytouseit.
- isaverypowerfulprograminitsownright.Ithandlesalargesubsetofthefunctionalityofthemostexpensiveandpowerful databasepackages.
- usesastandardformofthewell-knownSQLdatalanguage.
- worksonmanyoperatingsystemsandwithmanylanguagesincludingPHP,PERL,C,C++,JAVA, etc.
- worksveryquicklyandworkswellevenwithlargedatasets.
- isveryfriendlytoPHP,themostappreciatedlanguageforwebdevelopment.
- supportslargedatabases,upto50millionrowsormoreinatable.

  Thedefaultfilesizelimit foratableis4GB,butyoucanincreasethis(ifyouroperatingsystem canhandleit)toatheoretical limitof8millionterabytes(TB).
- iscustomizable.Theopen-sourceGPLlicenseallowsprogrammersto modifythe softwaretofittheirownspecificenvironments.

The SQLCREATETABLEStatement

TheCREATETABLEstatementisusedtocreateanewtableinadatabase.

Syntax

```
CREATETABLEtable_name(
  column1
  datatype,column2
  datatype,column3dataty
  pe,..............................);
```

Thecolumnparametersspecifythenamesofthecolumnsofthetable.

Thedatatypeparameterspecifiesthetypeofdatathecolumncanhold(e.g.varchar,integer,date,etc.).

SQLCREATETABLEExample

Thefollowingexamplecreatesatablecalled"Persons"thatcontainsfivecolumns:PersonID,LastName, FirstName,Address,and City:

Example

```
CREATETABLEPersons(P
    ersonIDint,
    LastName
    varchar(255),FirstName
    varchar(255),Address
    varchar(255),Cityvarcha
    r(255));
```

CreateTableUsingAnotherTable

Acopyofanexistingtable canbecreatedusinga combinationofthe CREATETABLEstatement andtheSELECTstatement.

Thenewtablegetsthesamecolumn definitions.Allcolumnsorspecificcolumnscanbeselected.

Ifyoucreateanewtableusinganexistingtable,thenewtablewill befilledwiththeexistingvaluesfromtheold table.

Syntax

```
CREATETABLEnew_table_nameASS
    ELECTcolumn1,column2,...
    FROMexisting_table_name
    WHERE.....;
```

SQLGeneralDataTypes

Eachcolumninadatabasetableisrequiredtohaveanameandadatatype.

SQL developers have to decide what types of data will be stored inside each and every tablecolumn when creating a SQL table. The data type is a label and a guideline for SQL tounderstand what type of data is expected inside of each column, and it also identifies howSQLwillinteractwith thestored data.

Thefollowingtableliststhegeneraldatatypesin SQL:

Datatype	**Description**
CHARACTER(n)	Characterstring. Fixed-lengthn
VARCHAR(n) orCHARACTER VARYING(n)	Characterstring.Variablelength.Maximumlengthn
BINARY(n)	Binarystring.Fixed-lengthn
BOOLEAN	StoresTRUEorFALSEvalues
VARBINARY(n) or BINARYVARYING(n)	Binarystring.Variablelength.Maximumlengthn

INTEGER(p)	Integernumerical (nodecimal).Precisionp
SMALLINT	Integernumerical (nodecimal).Precision5
INTEGER	Integernumerical (nodecimal).Precision10
BIGINT	Integernumerical (nodecimal).Precision19

DECIMAL(p,s)	Exact numerical, precision p, scale s. Example: decimal(5,2) is a numberthat has 3digits beforethe decimal and2digitsafter the decimal
NUMERIC(p,s)	Exactnumerical,precisionp,scale s. (Same as DECIMAL)
FLOAT(p)	Approximate numerical,mantissaprecisionp.Afloatingnumber inbase10exponential notation.The sizeargument forthis typeconsists of a singlenumberspecifyingtheminimumprecision
REAL	Approximate numerical,mantissaprecision 7
FLOAT	Approximate numerical,mantissaprecision16
DOUBLEP RECISION	Approximate numerical,mantissaprecision16
DATE	Storesyear,month,anddayvalues
TIME	Storeshour,minute,andsecondvalues
TIMESTAMP	Storesyear,month,day,hour,minute,andsecondvalues
INTERVAL	Composed of a number of integer fields, representing a period of time,dependingonthe type of interval
ARRAY	Aset-lengthand orderedcollection of elements
MULTISET	Avariable-length andunordered collection ofelements
XML	StoresXMLdata

TheSQLINSERTINTOStatement

TheINSERTINTOstatementisusedtoinsertnewrecordsinatable.

INSERTINTOSyntax

ItispossibletowritetheINSERTINTOstatementintwoways.

Thefirstwayspecifiesboththecolumnnamesandthevaluestobeinserted:INSER

T INTO*table_name*(*column1*,*column2*,*column3*,...)
VALUES(*value1*,*value2*,*value3*,...);

If you are adding values for all the columns of the table, you do not need to specify thecolumnnamesintheSQLquery.However,makesuretheorderofthevaluesisinthesameorderast hecolumnsinthetable.TheINSERTINTOsyntax wouldbeasfollows:

INSERTINTO*table_name*VALUES(*value1*,*value2*,*value3*,...);The

SELECTstatementisusedtoselectdatafromadatabase.

Thedatareturnedisstoredinaresulttable,calledtheresult-set.

SELECTSyntax

SELECT*column1*,*column2,...*FROM*table_name*;

Here,column1,column2,...arethefieldnamesofthetableyouwanttoselectdatafrom.Ifyouwanttos electallthefieldsavailablein thetable,usethefollowingsyntax:

SELECT*FROM*table_name*;

TheSQLAND,ORandNOTOperators

TheWHERE clausecanbecombinedwithAND,OR,andNOToperators.

TheANDandORoperatorsareusedtofilterrecords basedon morethan onecondition:

- TheANDoperatordisplaysa record ifalltheconditions separated byAND isTRUE.
- TheORoperatordisplaysarecordifanyoftheconditionsseparatedbyORisTRUE.

TheNOToperatordisplaysarecordifthecondition(s)isNOTTRUE.

ANDSyntax

```
SELECTcolumn1,column2,...FROMtable_name
WHEREcondition1ANDcondition2ANDcondition3 ...;
```

ORSyntax

```
SELECTcolumn1,column2,...FROMtable_name
WHEREcondition1OR condition2ORcondition3 ...;
```

NOTSyntax

```
SELECTcolumn1,column2,...FROMtable_name
WHERENOTcondition;
```

AND Example : The following SQL statement selects all fields from "Customers" wherecountryis "Germany"ANDcityis"Berlin":

Example

```
SELECT*FROMCustomers
WHERECountry='Germany'ANDCity='Berlin';
```

OR Example : The following SQL statement selects all fields from "Customers" where cityis"Berlin"OR"München":

Example

```
SELECT*FROMCustomers
WHERECity='Berlin'ORCity='München';
```

NOT Example : The following SQL statement selects all fields from "Customers" wherecountryis NOT"Germany":

Example

```
SELECT * FROM
CustomersWHERENOTCountry='
Germany';
```

TheSQLORDER BYKeyword

TheORDERBYkeywordisusedtosorttheresult-setinascendingordescendingorder.

TheORDERBYkeywordsortstherecordsinascendingorderbydefault.Tosorttherecordsi ndescendingorder,usetheDESCkeyword.

ORDERBYSyntax

```
SELECTcolumn1,column2,...
FROMtable_name
ORDERBYcolumn1,column2,...ASC|DESC;
```

ORDERBYExample

The following SQL statement selects all customers from the "Customers" table, sorted bythe"Country"column:

Example

```
SELECT*FROMCustomersORDERBYCountry;
```

TheSQLSELECTDISTINCTStatement

TheSELECTDISTINCTstatementisusedtoreturnonlydistinct(different)values.Insideatable, a column often contains many duplicate values; and sometimes you only want to listthe different (distinct) values. The SELECT DISTINCT statement is used to return onlydistinct (different)values.

SELECTDISTINCTSyntax

```
SELECTDISTINCTcolumn1,column2,...FROMtable_name;
```

SELECTDISTINCTExamples

The followingSQLstatementselectsonlytheDISTINCTvalues fromthe"Country"columninthe"Customers"table:

Example

```
SELECTDISTINCTCountryFROMCustomers;
```

TheSQLWHEREClause

TheWHERE clauseisusedtofilterrecords.

TheWHEREclauseisusedtoextract onlythoserecordsthatfulfill aspecifiedcondition.

WHERE Syntax

```
SELECTcolumn1,column2,..FROMtable_nameWHEREcondition;
```

WHEREClauseExample

The followingSQLstatementselects allthecustomersfromthecountry"Mexico", in the"Customers"table:

Example

```
SELECT*FROMCustomersWHERECountry='Mexico';
```

TextFieldsvs.NumericFields

SQL requires single quotes around text values (most database systems will also allow doublequotes).

However,numericfieldsshouldnotbeenclosedinquotes:

Example

```
SELECT*FROMCustomersWHERECustomerID=1;
```

OperatorsinTheWHEREClause

Thefollowingoperatorscanbeusedinthe WHERE clause:

Operator	Description
=	Equal
<>	Not equal. **Note:**InsomeversionsofSQLthis operatormaybewrittenas!=
>	Greaterthan
<	Less than
>=	Greater than or equal
<=	Less thanorequal
BETWEEN	Betweenaninclusive range
LIKE	Searchforapattern
IN	Tospecifymultiplepossiblevaluesforacolumn

TheSQLDELETEStatement

TheDELETEstatementisusedtodeleteexistingrecordsinatable.

DELETESyntax

```
DELETEFROMtable_nameWHEREcondition;
```

Note: Be careful when deleting records in a table! Notice the WHERE clause in theDELETEstatement.TheWHEREclausespecifieswhichrecord(s)thatshouldbedeleted.Ifyou omittheWHERE clause,allrecordsin thetablewillbedeleted!

SQLDELETEExample

The following SQL statement deletes the customer "Alfreds Futterkiste" from the"Customers"table:

Example

```
DELETEFROMCustomers
WHERECustomerName='AlfredsFutterkiste';
```

DeleteAllRecords

Itispossibletodeleteallrowsinatablewithoutdeletingthetable.Thismeansthat thetablestructure,attributes, and indexes willbeintact:

```
DELETEFROMtable_name;
```

TheSQLMIN() and MAX() Functions

TheMIN()functionreturnsthesmallestvalueoftheselectedcolumn.The

MAX() function returns the largest value of the selected

column.**MIN()Syntax**

```
SELECT
MIN(column_name)FROM
table_name
WHEREcondition;
```

MAX()Syntax

```
SELECTMAX(column_name)FROMtable_nameWHEREcondition;
```

MIN()Example

The followingSQLstatementfindstheprice ofthe cheapestproduct:

Example

```
SELECTMIN(Price)ASSmallestPriceFROMProducts;
```

MAX()Example

The followingSQLstatementfinds thepriceofthemostexpensiveproduct:

Example

```
SELECTMAX(Price)ASLargestPriceFROMProducts;
```

TheSQLCOUNT(),AVG() andSUM() Functions

TheCOUNT()functionreturnsthenumberofrowsthatmatchesaspecifiedcriteria.TheA

VG() functionreturnsthe average value of anumeric column.

TheSUM()functionreturnsthetotal sum ofanumericcolumn.

COUNT()Syntax

```
SELECTCOUNT(column_name)FROMtable_name
WHEREcondition;
```

AVG()Syntax

```
SELECTAVG(column_name)FROMtable_name
WHEREcondition;
```

SUM()Syntax

```
SELECTSUM(column_name)FROMtable_name
WHEREcondition;
```

COUNT()Example

The followingSQLstatementfinds thenumber of products:

Example

```
SELECTCOUNT(ProductID)FROMProducts;
```

AVG()Example

The followingSQLstatementfindstheaverage priceofall products:

Example

```
SELECTAVG(Price)FROMProducts;
```

SUM()Example

The following SQL statement finds the sum of the "Quantity" fields in the "OrderDetails"table:

Example

```
SELECTSUM(Quantity)FROMOrderDetails;
```

TheSQLLIKE Operator

TheLIKEoperatorisusedinaWHEREclausetosearchforaspecifiedpatterninacolumn.Therearet

wowildcardsusedin conjunctionwiththeLIKEoperator:

- %-Thepercentsignrepresentszero,one,ormultiplecharacters
- _-Theunderscorerepresentsasinglecharacter

Thepercent (%)signandtheunderscore(_)canalsobeusedincombinations!

LIKESyntax

```
SELECTcolumn1,column2,...FROMtable_name
WHEREcolumnNLIKEpattern;
```

Tip: You can also combine any number of conditions using AND or OR

operators.HerearesomeexamplesshowingdifferentLIKEoperatorswith'%'and'_'wildcar

ds:

LIKEOperator	Description
WHERECustomerNameLIKE'a%'	Finds anyvaluesthatstarts with "a"
WHERECustomerNameLIKE'%a'	Finds anyvalues that ends with"a"
WHERE CustomerName LIKE'%or%'	Finds anyvalues thathave "or"in anyposition
WHERE CustomerName LIKE'_r%'	Finds anyvaluesthat have "r"inthe second position
WHERECustomerNameLIKE 'a_%_%'	Finds anyvaluesthatstartswith "a"and areatleast 3 charactersin length
WHEREContactNameLIKE'a%o'	Finds anyvaluesthat startswith "a"andendswith "o"

SQLLIKEExamples

The followingSQLstatementselectsallcustomerswithaCustomerNamestartingwith "a":

Example

```
SELECT * FROM
CustomersWHERECustomerNameL
IKE'a%';
```

The followingSQLstatementselectsallcustomers withaCustomerNameendingwith "a":

Example

```
SELECT * FROM
CustomersWHERECustomerNameL
IKE'%a';
```

The following SQL statement selects all customers with a CustomerName that have "or" inanyposition:

Example

```
SELECT*FROMCustomers
WHERECustomerNameLIKE'%or%';
```

The following SQL statement selects all customers with a CustomerName that have "r" inthesecond position:

Example

```
SELECT * FROM
```

```
CustomersWHERECustomerNameLI
KE'_r%';
```

The following SQL statement selects all customers with a Customer Name that starts with"a"andareatleast 3 charactersin length:

Example

```
SELECT*FROMCustomers
WHERECustomerNameLIKE'a_%_%';
```

The following SQL statement selects all customers with a Customer Name that starts with"a"andends with "o":

Example

```
SELECT * FROM
CustomersWHEREContactNameLI
KE'a%o';
```

The following SQL statement selects all customers with a CustomerName that NOT startswith"a":

Example

```
SELECT*FROMCustomers
WHERECustomerNameNOTLIKE'a%';
```

Conclusion: Thuswehavestudiedhowtouseopensourcedatabase.

FAQ?

1. CompareVs.SQLServer.
2. Whatarethefeaturesof?
3. What doDDL,DML,andDCLstand for?
4. WhatisthedifferencebetweenCHARandVARCHAR?
5. Whatisthedifferencebetweenprimarykeyandcandidatekey?
6. WhatisthedifferencebetweenDELETETABLEandTRUNCATETABLE&DROPtablecommands in?

AssignmentNo.	**2**
Title	**SQLQueries:**
PROBLEM STATEMENT/DEFINITION	a. Design and Develop SQLDDL statements which demonstrate the use of SQL objects suchasTable,View,Index,Sequence,Synonym,differentconstraintsetc. b. Writeatleast10SQLqueriesonthesuitabledatabaseapplicationusingSQLDMLstatements
Objectives	- Understand & implement the various DDL Commands. - Understand database concepts like view, index ,sequence and synonym
Software packages and hardware apparatus used	MySQL/Oracle PC with the configuration as Latest Version of 64 bit Operating Systems, Open Source Fedora-GHz. 8 G.B. RAM, 500 G.B. HDD, 15"Color Monitor, Keyboard, Mouse
References	Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", 5th Edition, McGraw Hill Publishers, 2002, ISBN 0-07-120413-X Connally T., Begg C., "Database Systems", 3rd Edition, Pearson Education, 2002, ISBN 81-7808-861-4 mysql.com/docs/mysql-tutorial-excerpt-5.1-en.pdf
STEPS	Refer to details
Instructions for writing journal	- Date - Title - Problem Definition - Learning Objective - Learning Outcome - Theory-Related concept,Architecture,Syntax etc - Class Diagram/ER diagram - Test cases - Program Listing - Output - Conclusion

AssignmentNo.2

Title: SQL Queries

a. Design and Develop SQLDDL statements which demonstrate the use of SQL objects suchasTable,View,Index,Sequence,Synonym,differentconstraintsetc.

b. Writeatleast10SQLqueriesonthesuitabledatabaseapplicationusingSQLDMLstatements

Objectives: Understand & implement the various DDL Commands.
Understand database concepts like view, index ,sequence and synonym

Theory:SQL–StructuredQueryLanguage

Data Definition in

SQLCreating

TablesSyntax:-

Create table<table

name>(colume_name 1 datatype

size(),colume_name2datatypesiz

e(),

….

colume_namendatatypesize());

e.g. Createtablestudentwiththefollowingfields(name,roll,class,branch)
Createtablestudent
(name

char(20),Roll

number(5),Class

char(10),Branchc

har(15));

<u>Atablefroma table</u>

- **Syntax :**

CREATE TABLE <TableName> (<ColumnName>, <Columnname>) ASSELECT<ColumnName>,<Columnname>FROM<TableName>;

-
Ifthesourcetablecontainstherecords,thennewtableisalsocreatedwiththesamerecordspresentin thesourcetable.

- Ifyouwantonlystructurewithoutrecordsthenselectstatementmusthavecondition.Syntax:

CREATE TABLE <TableName> (<ColumnName>, <Columnname>)ASSELECT <ColumnName>, <Columnname> FROM

<TableName> WHERE1=2; (Or)

CREATE TABLE <TableName> (<ColumnName>, <Columnname>)ASSELECT <ColumnName>, <Columnname> FROM <TableName> WHEREColumnName=NULL;

Constraints

Thedefinitionofatablemayincludethespecificationofintegrityconstraints.Basicallytwo types of constraints are provided: column constraints are associated with a singlecolumn whereas table constraints are typically associated with more than one column. Aconstraint can be named. It is advisable to name a constraint in order to get moremeaningful information when this constraint is violated due to, e.g., an insertion of atuple that violates the constraint. If no name is specified for the constraint, Oracleautomatically generates a name of the pattern SYS C<number>.Rules are enforced ondatabeingstored in atable, arecalled **Constraints**.

Boththe Create table &AlterTable SQLcanbe usedtowrite SQLsentences thatattachconstraints.

Basicallyconstraintsareofthreetypes

1) Domain
   - NotNull
   - Check
2) Entity
   - PrimaryKey
   - Unique
3) Referential
   - Foreignkey
4) NotNull:-Notnullconstraintcanbeappliedatcolumnlevelonly.

Wecandefinetheseconstraints

1) atthetimeoftablecreationSyntax:

CREATETABLE<tableName>(<ColumnName>datatype(size)NOTNULL,

<ColumnName>datatype(size),….

);

2) Afterthetablecreation

ALTER TABLE <tableName> Modify(<ColumnName>datatype(size)NOTNULL);

Checkconstraints

- Can be bound to column or a table using CREATE TABLE or ALTER TABLEcommand.

- Checksareperformedwhenwriteoperationisperformed.
- Insertorupdatestatementcausestherelevantcheckconstraint.
- Ensurestheintegrityofthedataintables.

Syntax:

- CheckconstraintsatcolumnlevelS

yntax :

CREATE TABLE

<tableName>(<ColumnName>datatype(size)CHECK(columnNameconditi

on),<columnnamedatatype(size));

CREATETABLE<tableName>

(<ColumnName> datatype(size)CONSTRAINT<constraint_name>

CHECK(columnNamecondition),..

);

- CheckconstraintsattablelevelS

yntax :

CREATETABLE<tableName>

(<ColumnName>datatype(size),
<ColumnName>datatype(size),
CONSTRAINT<constraint_name>CHECK(columnNamecondition),..);

- Checkconstraintsattablelevel

Syntax:

CREATETABLE

<tableName>(<C

olumnName>dat

atype(size),

<ColumnName>datatype(size),….,

CHECK(columnNamecondition));

Aftertablecreation

Altertabletablename

Addconstraintsconstraintnameckeck(condition)T

hePRIMARYKEYConstraint

Aprimarykeyisoneormore column(s) in atableused to uniquelyidentifyeach row inthetable.

- Atable can have onlyoneprimarykey.
- Cannot beleft blank

- DatamustbeUNIQUE.
- Not allowsnullvalues
- Notallowsduplicatevalues.
- Uniqueindexiscreatedautomaticallyifthereisaprimarykey.Prim

arykeyconstraintdefinedatcolumnlevel

Syntax:

CREATETABLE<TableName>

(<ColumnName1><DataType>(<Size>)PRIMARY KEY,<columnname2

<datatype(<size>),…..);

- Primarykeyconstraintdefinedat TablelevelSy

ntax:

CREATETABLE<TableName>

(<ColumnName1> <DataType>(<Size>) PRIMARY ,…,

KEY(<ColumnName1><ColumnName2>));

- keyconstraintdefinedatTablelevel

Syntax:

CREATETABLE<TableName>

(<ColumnName1> <DataType>(<Size>) <columnname2datatype<(size)<,<columnname3

datatype<size>constraint

constraintnamePRIMARYKEY(<ColumnName1>));

Aftertablecreation

Altertabletablename

Add(constraintconstraintnameprimarykey(columnname));

TheUniqueKeyConstraint

- Theuniquecolumnconstraintpermitsmultipleentriesof NULLintothecolumn.
- Uniquekeynotallowedduplicatevalues
- Uniqueindexisautomaticallycreated.
- Tablecanhavemorethanoneuniquekey.

- UNIQUEconstraintdefinedatcolumnlevelSy
ntax :

Createtabletablename(**<columnname><datatype>(<Size>**

UNIQUE),<columnname>datatype(<size>)................);

UNIQUEconstraintdefinedattablelevelSy
ntax :

CREATE TABLE tablename (<columnname> <datatype>(<Size>), <columnname> <datatype>(<Size>), UNIQUE(<columnname>, <columnname>));

Aftertablecreation

Altertabletablename

Addconstraintconstraintnameunique(columnname);

- TheForeignKey(SelfReference)Constraint

  Foreignkeyrepresentsrelationshipsbetweentables.

Aforeignkeyisa column(orgroupofcolumns) whosevalues arederivedfromprimarykeyor uniquekeyofsomeother table.

ForeignkeyconstraintdefinedatcolumnlevelSy

ntax:

<columnName><DataType>(<size>)REFERENCES<TableName>[(<ColumnName>)] [ONDELETECASCADE]

- IftheONDELETECASCADEoptionisset,aDELETEoperationinthemaster table will trigger a DELETE operation for corresponding recordsinalldetailtables.
- If the ON DELETE SET NULL option is set, a DELETE operation in themaster table will set the value held by the foreign key of the detail tablesto null.

Foreignkey:

ALTER TABLE <child_tablename>ADD CONSTRAINT <constraint_name> FOREIGNKEY(<columnnameinchild_table>)REFERENCES<parenttablename>;

1) FOREIGNKEYconstraintattablelevel
2) FOREIGN KEY constraint defined with ON DELETE CASCADEFOREIGNKEY(<ColumnName>[,<columnname>])REFERENCES
   <TableName>[(<ColumnName>,<ColumnName>)ONDELETE
   CASCADE

- FOREIGNKEYconstraintdefinedwithONDELETESETNULL

FOREIGNKEY(<ColumnName>[,<columnname>])REFERENCES <TableName>[(<ColumnName>,<ColumnName>)ONDELETESETNULL

- To view

theconstraintSyntax:

Select constraint_name, constraint_type,search_condition fromuser_constraintswheretable_name=<tablename>;
Selectconstraint_name,column_namefromuser_cons_columnswheretable_name=<tablename>;

TodroptheconstraintsS

yntax:-

Dropconstraintconstraintname;

Describecommands

Toviewthestructureofthetablecreatedusethe**DESCRIBE**command.Thecommanddisplaysthecolumnnames anddatatypes

Syntax:-

Desc[ribe]<table_name>

e.gdescstudent

Restrictionsforcreatingatable:

1.Table names and column names must begin with a letter.2.Tablenamesandcolumnnamescanbe1to30characterslong.

3.table names must contain only the characters A-Z,a-z,0-9,underscore_,$ and #4.Tablenameshouldnot besameasthenameofanotherdatabaseobject.

5.TablenamemustnotbeanORACLEreservedword.

6.Columnnamesshouldnot beduplicatewithinatabledefinition.

AlterationofTABLE:-

Alter table

commandSyntax:-

Case1:-

Altertable<table_name>

Add(colume_name 1 datatype size(),colume_name2datatypesize(),

colume_namendatatypesize());

Case2:-

```
Alter table
<table_name>Modify(colume_name1da
tatypesize(),
                colume_name2datatypesize(),
                 …...
                 colume_namendatatypesize());
```

Afteryoucreateatable,youmayneedtochangethetablestructuresbecauseyouneedtohaveacolumnd efinition needstobechanged.Altertablestatementcanbeused forthispurpose.
Youcanadd columns toa tableusingthe altertablestatementwiththeADDclause.

E.g.Supposeyouwanttoaddenroll_nointhestudenttablethenwewrite
AltertablestudentAdd(enroll_nonumber(10));

Youcanmodifyexistingcolumninatablebyusingthealtertablestatementwithmodifyclause.

E.g.Supposeyouwant tomodifyorchangthesizeofpreviouslydefinedfieldnameinthestudenttablethen wewrite

Altertablestudentmodify(namechar(25));D roppinga columnfromatable

Syntax:

ALTERTABLE<Tablename>DROPCOLUMN <ColumnName>;

DroptablecommandSyntax:-

Droptable<table_name>

Droptablecommandremnovesthedefinitionsofanoracletable.Whenyoudropatable ,thedatabaselosesall thedatainthetableandalltheindexesassociatedwithit.

e.gdroptablestudent;

Truncatetablecommand
Syntax:-

Trunctable<table_name>

Thetruncatetablestatementisusedtoremoveallrowsfromatableandtoreleasethestoragespaceused bythetable.

e.g.Trunctablestudent;

Renametablecommand

Syntax:-

Rename<oldtable_name> to<newtable_name>

Renamestatementisusedtorenameatable,view,sequence,orsynonym.e.g.Renamestudenttostud;

Database objects:-

Index

An index is a schema object that can speed up retrieval of rows by using pointer. An indexprovidesdirect&fastaccesstorowsinatable.Indexcanbecreatedexplicitlyorautomatically.

Automatically:-Auniqueindexiscreated automaticallywhen youdefineaprimarykeyoruniquekeyconstraintin atabledefinition.

Manually:-userscancreatenonuniqueindexesorcolumnstospeedupaccesstimetotherows.

Syntax:

Create index<index_name>On table(column[, column] …);Eg. Createindex emp_ename_idx On emp(ename);

Whentocreateanindex

a) ThecolumnisusedfrequentlyintheWHEREclauseor inajoincondition.

b) Thecolumncontainsawiderangeofvalues.

c) Thecolumncontainsalargenumberofvalues.

Todisplaycreatedindexofatableeg.

Selectic.index_name,ic.column_name,ic.colun_positioncol_pos,ix.uniquenessfrom user_indexes ix, user_ind_columns ic where ic.index_name=ix.index_nameand ic.table_name=‟emp‟;

RemovinganIndex

Syntax:-

Dropindex<index_name>;eg.Dropin dexemp_name_idx;

Note:1)wecannotmodifyindexes.

2)Tochangeanindex,wemustdropitandthere-createit.

Views

View is a logical representation of subsets of data from one or more tables. A view takes theoutputofaqueryandtreatsitasatablethereforeviewcanbecalledasstoredqueryoravirtualtable.The tables upon which a view is based are called base tables. In Oracle the SQLcommandto createaview(virtualtable)has theform

Create[orreplace]view<view-name>[(<column(s)>)]as

<select-statement>[withcheckoption[constraint<name>]];

The optional clause or replace re-creates the view if it already exists. <column(s)> names thecolumnsoftheview.If<column(s)>isnotspecifiedintheviewdefinition,thecolumnsoftheviewg et thesamenamesastheattributes listedintheselectstatement(ifpossible).

Example:Thefollowingviewcontainsthename,jobtitleandtheannualsalaryofemployeesworking in thedepartment20:

CreateviewDEPT20as
selectENAME,JOB,SAL12ANNUALSALARYfromEMPwhereDEPTNO =20;

Inthe selectstatementthecolumnaliasANNUALSALARYis specifiedfor theexpressionSAL*12and this aliasis taken bytheview.Analternativeformulationoftheaboveview definitionis

Create viewDEPT20(ENAME, JOB,ANNUALSALARY) asselectENAME,JOB, SAL fromEMPwhereDEPTNO=20;

A view can be used in the same way as a table, that is, rows can be retrieved from a view(alsorespectiverowsarenotphysicallystored,butderivedonbasisoftheselectstatementintheviewdefinition), or rows can even be modified. A view is evaluated again each time it is accessed. InOracle SQLnoinsert,update, or deletemodifications onviews areallowed that useoneofthefollowingconstructsintheviewdefinition:

- Joins
- Aggregatefunctionsuchassum,min,maxetc.
- set-valuedsubqueries(in,any,all)ortestforexistence(exists)
- groupbyclauseordistinctclause

In combination with the clause with check option any update or insertion of a row into the viewis rejected if the new/modified row does not meet the view definition, i.e., these rows would notbe selectedbasedon theselect statement.Awith checkoption canbenamed usingtheconstraintclause.

A view can be deleted using the command delete <view-name>. Todescribethestructureofaview

e.g.**Describestud;**

Todisplaythecontentsofviewe.g.Select*fromstud

Removingaview:

Syntax:-**Dropview<view_name>**
e.g.Dropviewstud

Sequence:

A sequence is a database object, which can generate unique, sequential integer values. It can beusedtoautomaticallygenerate primarykeyor uniquekeyvalues.Asequence can beeitherin anascendingor descendingorder.

Syntax:

```
Create
sequence<sequence_name>
[incrementbyn]
[start withn]
[{maxvalue n |
nomaxvalue}] [{minvalue n|
nominvalue}] [{cycle |
nocycle}]
[{cachen|nocache}];
```

Incrementbyn	Specifies the interval betweensequencenumberwherenis aninteger.Ifthisclauseisomitted,the sequenceisincrementby1.
Startwithn	Specifies the first sequence number tobegenerated.Ifthisclauseisomitted, thesequenceisstartwith1.
Maxvaluen	Specifiesthemaximumvalue,the sequencecangenerate
Nomaxvaluen	Specifiesthemaximumvalueof10e27-1foranascendingsequence&-1fordescendingsequence.Thisisa defaultoption.
Minvaluen	Specifies the minimum sequence value.
Nominvaluen	Specifies the minimum value of 1 foranascending&10e26-1fordescendingsequence.Thisisadefault option.
Cycle	Specifiesthatthesequencecontinues togeneratevaluesfromthebeginning afterreachingeitheritsmaxorminvalue.
Nocycle	Specifies that the sequence can notgenerate more values after reachingeitheritsmaxorminvalue.Thisisa defaultoption.
Cache/nocache	Specifieshowmanyvaluestheoracleserverwillpreallocate& keep

	inmemory.Bydefault,theoracleserver willcache20values.

Aftercreatingasequencewecanaccessitsvalueswiththehelpofpseudocolumnslike**cur val**&**nextval**.

Nextval:nextvalreturnsinitialvalueofthesequencewhenreferencetoforthefirsttime. Last references to the nextval will increment the sequence using the incrementbyclause&returns thenewvalue.

Curval:curvalreturnsthecurrentvalueofthesequencewhichisthevaluereturnedbythelas treferenceto lastvalue.

Modifyningasequence:
Thesequencecanbemodifiedwhenwewanttoperformthefollowing:

□ Setoreliminate minvalueormaxvalue

□ Changetheincrementvalue.

Change thenumberofcachesequencenumber.

Syntax:

```
Alter sequence <sequence_name>
[incrementbyn]
[start withn]
[{maxvaluen|nomaxvalue}]
[{minvaluen|nominvalue}]
[{cycle|nocycle}]
[{cachen|nocache}];
```

Synonym:

A synonym is a database object,which is used as an alias(alternative name)for atable,vieworsequence.

Syntax:-

```
Create[public]synonym
<synonym_name>for<table_name>;
```

Inthesyntax

Public:-Creates a synonym accessible to all
users.Synonym:-
Isthenameofthesynonymtobecreated.

Synonym can either be private or public.A private synonym is created by normaluser,whichis availabletothatpersons.

A public synonym is created by a database administrator(DBA),which can be availed byanyotherdatabaseuser.

Uses:-

1.SimplifySQLstatements.

2.Hidethenameandownerofanobject.3.Pr
ovidepublicaccesstoanobject.
Guidelines:-

1.User can do all DML manipulations such as insert ,delete,update on synonym.2.User cannot perform any DDL operations on the synonym except dropping thesynonym.

3.Allthemanipulationsonitactuallyaffectthetablee.gCreat
esynonymstud1 forstudent;

SQL, pronounced SEQUEL, is the standard language to access relational databases.SQL is anabbreviation for Structured Query Language. I'll just add that SQL is composed of DML andDDL.DMLarethekeywordsyouusetoaccessandmanipulatedata,hencethenameDataManipulation Language. DDL arethe keywords you use to create objectssuch as views,tablesand procedures, hencethenameDataDefinitionLanguage.

Tables
Inrelationaldatabase systems(DBS)data arerepresentedusingtables (relations).Aqueryissuedagainstthe DBSalsoresultsin atable.Atablehas thefollowingstructure:

Attributes	Column1	Column2	. . .	Column n
	←Tuple(or	Record)		
	……….	……….	……	………..

Atableisuniquelyidentifiedbyitsnameandconsistsofrowsthatcontainthestoredinformation, each row containing exactly one tuple (or record). A table can have one or morecolumns.

A column is made up of a column name and a data type, and it describes an attribute of thetuples. The structure of a table, also called relation schema, thus is defined by its attributes.Thetypeofinformationtobestoredinatableisdefinedbythedatatypesofthe attributesattable creationtime.SQLuses theterms
table,row,andcolumnforrelation,tuple,andattribute,respectively.

Atable canhave up to 254 columnswhich mayhavedifferent eventorsamedatatypesandsetsof values(domains),respectively.Possibledomainsarealphanumericdata(strings),numbersanddateformats.

Datatype	**Description**	**MaxSize: Oracle7**	**MaxSize:Oracle8**	**MaxSize: Oracle9**	**MaxSize:PL/SQL**	**PL/SQL Subtypes/ Synonyms**
VARCHAR2(size)	Variablelength characterstringhavingmaximum length *size*bytes. You mustspecify size	2000 bytesminimumis1	**4000** bytesminimumis 1	**4000**bytes minimum is 1	32767 bytesminimumis1	STRING VARCHAR
NVARCHAR2 (size)	Variablelength nationalcharacter setstringhavingmaximumlength *size*bytes. You mustspecify size	N/A	4000 bytesminimumis 1	4000bytes minimum is 1	32767 bytesminimumis1	STRING VARCHAR
VARCHAR	Nowdeprecated -VARCHARis asynonymfor VARCHAR2 but this usagemaychange in futureversions.	-	-	-		
CHAR(size)	Fixed lengthcharacterdataoflengthsizebytes. Thisshould be usedfor fixed lengthdata. Such ascodes A100,B102...	255 bytesDefaultandminimumsize is 1byte.	**2000** bytesDefaultandminimum size is1byte.	**2000** bytesDefault andminimum sizeis1byte.	32767 bytesDefaultandminimumsize is 1byte.	CHARACTER
NCHAR (size)	Fixedlength nationalcharacterset	N/A	2000 bytesDefault	2000bytes Default andminimumsize	32767 bytesDefault	

	data of lengthsize bytes. Thisshould be usedforfixedlengthdata. Such ascodes A100,B102...		andminimum size is1byte.	is1byte.	andminimumsize is 1byte.	
NUMBER(p,s)	Number havingprecision p andscales.	Theprecisionp canrangefrom1to 38. The scales canrangefrom-84 to 127.	Theprecision p canrangefrom 1 to38. Thescale scanrangefrom-84 to127.	The precisionp can rangefrom 1to38. The scale scan rangefrom-84to 127.	Magnitude 1E-130.. 10E125 maximum precisionof 126binary digits,which isroughlyequivalent to 38decimal digits The scales canrangefrom-84 to 127.For floating pointdon'tspecifyp,sREAL has amaximumprecisionof 63binarydigits,which isroughlyequivalent to 18decimal digits	fixed-pointnumbers:DECDECIMALNUMERIC floating-point:DOUBLEPRECISIONFLOAT binary_doublebinary_float integers:INTEGERINTSMALLINT simple_integer(10g) BOOLEAN REAL
PLS_INTEGER	signedintegers PLS_INTEGER values requirelessstorageand	PL/SQL only	PL/SQL only	PL/SQLonly	magnituderangeis -21474836	

	provide betterperformancethan NUMBERvalues. So usePLS_INTEGER whereyoucan!				47 .. 2147483647	
BINARY_INTEGER	signed integers(older slowerversion ofPLS_INTEGER)				magnituderangeis -2147483647 .. 2147483647	NATURALNATURALNPOSITIVEPOSITIVENSIGNTYPE
LONG	Character dataof variablelength (A biggerversion theVARCHAR2 datatype)	2 Gigabytes	2 Gigabytes	2Gigabytes- but nowdeprecated	32760 bytesNote thisissmallerthan themaximumwidth of aLONG column	
DATE	Validdaterange	fromJanuary1, 4712BC toDecember31, 4712AD.	fromJanuary1, 4712 BC toDecember31, **9999** AD.	from January1,4712 BCto December31, **9999**AD.	fromJanuary1, 4712BC toDecember31, **9999**AD. (inOracle7 =4712AD)	
TIMESTAMP (fractional_seconds_precision)	the number ofdigits in thefractional part ofthe SECONDdatetimefield.	-	-	Accepted values offractional_seconds_precisionare0 to9. (default=6)		
TIMESTAMP (fractional_seconds_precision)WITH {LOCAL}TIMEZONE	Asabovewithtime zonedisplacementvalue	-	-	Acceptedvalues offractional_seconds_precisionare0 to9. (default=6)		

YEAR (year_precision)TOMONTH	and months,whereyear_precisionisthenumberofdigits in theYEAR datetimefield.			valuesare0to 9.(default=2)		
INTERVAL DAY (day_precision) TOSECOND (fractional_seconds_precision)	Timeindays, hours,minutes,andseconds. *day_precision* isthe maximumnumberofdigitsin'DAY' *fractional_seconds_precision* isthe max numberof fractionaldigits in theSECONDfield.	-	-	*day_precision* maybe0 to 9. (default=2) *fractional_seconds_precision*maybe0to 9. (default =6)		
RAW(size)	Rawbinarydata oflengthsizebytes. You mustspecifysizeforaRAWvalue.	Maximumsize is 255bytes.	Maximum size is**2000** bytes	Maximumsize is **2000**bytes	32767 bytes	
LONGRAW	Rawbinarydataof variablelength. (notintrepreted byPL/SQL)	2 Gigabytes.	2 Gigabytes.	2Gigabytes- but nowdeprecated	32760 bytesNote thisissmallerthan themaximumwidth of aLONGRAW column	
ROWID	Hexadecimalstringrepresentingtheunique addressof a row in itstable. (primarily forvalues returnedby the ROWIDpseudocolumn.)	8bytes	10bytes	10bytes	Hexadecimal stringrepresenting theunique addressofa row inits table. (primarily forvalues returned	

					by theROW ID pseudocolumn.)	
UROWID	Hex stringrepresenting thelogical addressof a row of anindex-organizedtable	N/A	Themaximum sizeanddefaultis 4000bytes	The maximumsize anddefaultis4000bytes	universal rowid - Hexstringrepresenting thelogical addressofa row ofan index-organizedtable,either physical,logical, orforeign(non-Oracle)	SeeCHARTOROWID and thepackage:DBMS_ROWID
MLSLABEL	Binaryformatofan operatingsystemlabel.Thisdatatype is usedwith TrustedOracle7.					
CLOB	CharacterLarge Object	4Gigabytes	4Gigabytes	4Gigabytes	4Gigabytes	
NCLOB	NationalCharacter LargeObject		4Gigabytes	4Gigabytes	4Gigabytes	
BLOB	Binary LargeObject		4Gigabytes	4Gigabytes	4Gigabytes	
BFILE	pointertobinaryfileon disk		4Gigabytes	4Gigabytes	Thesize of aBFILEis systemdependentbut cannotexceedfourgigabytes(2**32 - 1bytes).	

XMLType	XMLdata	-	-	4Gigabytes	Populate withXML from aCLOB orVARCHAR2. or queryfromanother XMLTypecolumn.	

FAQ: Considerrelationalschema**Student(Roll_no,Name,Deptno,Marks,Email_id)** DevelopSQLDDLstatements.

1. CreatetableStudent;
2. Insertvaluesinstudenttable.
3. Addanewattributedateofbirthinstudentrecordusingalterstatement.
4. Dropdateofbirthattributefrom studenttable.
5. Updateastudentmarkswhererollnois7;
6. Deletearecordofstudentwhoserollnois4;
7. Createviewforstudenttable;
8. CreateindexonRollnoinstudenttable.
9. Createsequenceonstudenttable.
10. Createsynonymonstudenttable.

□FetchingDatafromTable:

The SQL**SELECT**commandisusedtofetchdata fromdatabase.Youcan usethiscommand at >promptaswellasinanyscriptlikePHP.Synt
ax:
Here isgenericSQLsyntaxofSELECTcommandtofetch datafromtable:

```
SELECTfield1,field2,...fieldNtable_name1,table_name2...
[WHEREClause]
[OFFSETM][LIMITN]
```

- □ YoucanuseoneormoretablesseparatedbycommatoincludevariousconditionsusingaWHERE clause,butWHERE clauseisanoptionalpartofSELECTcommand.
- □ YoucanfetchoneormorefieldsinasingleSELECTcommand.
- □ Youcanspecifystar(*)inplaceoffields.Inthiscase,SELECTwillreturnallthefields.
- □ YoucanspecifyanyconditionusingWHEREclause.
- □ Youcanspecifyanoffsetusing**OFFSET**fromwhereSELECTwillstartreturningrecords. Bydefaultoffset is zero.
- □ Youcanlimitthenumberofreturnsusing**LIMIT**attribute.

FetchingDatafromCommandPrompt:
This will use SQL SELECT command to fetch data fromtable tutorials_tblExample:
Followingexamplewillreturnalltherecordsfrom**tutorials_tbl**table:

```
>SELECT*fromtutorials_tbl
+++++
|tutorial_id|tutorial_title|tutorial_author|submission_date|
+++++
|        1|LearnPHP        |JohnPoul       |2007-05-21|
|        2|Learn        |Abdul S        |2007-05-21      |
|        3|JAVATutorial|Sanjay              |2007-05-21      |
+++++
```

The SQL**SELECT**statement returns aresult set of records from one or more tables.A**SELECT**statementretrieveszeroormorerowsfromoneormoredatabasetablesordatabase views. In most applications, SELECT is the most commonly used Data ManipulationLanguage (DML) command. As SQL is a declarative programming language, SELECT queriesspecify a result set, but do not specify how to calculate it. The database translates the query into a"query plan" which may vary between executions, database versions and database software. Thisfunctionality is called the "query optimizer" as it is responsible for finding the best possibleexecutionplanforthequery, within applicableconstraints.
TheSELECTstatementhasmanyoptionalclauses:

- □ WHEREspecifieswhichrowstoretrieve.

- GROUPBYgroupsrowssharingapropertysothatanaggregatefunctioncanbeappliedtoeach group.
- HAVINGselectsamongthegroupsdefinedbytheGROUPBYclause.
- ORDERBYspecifiesanorderinwhichtoreturntherows.
- ASprovidesanaliaswhichcanbeusedtotemporarilyrenametablesorcolumns.

WHEREClause

We have seen SQL **SELECT** command to fetch data fromtable. We can use a conditionalclause called **WHERE** clause to filter out results. Using WHERE clause, we can specify aselectioncriteriato selectrequired records fromatable.

Syntax:

Here is generic SQL syntax of SELECT command with WHERE clause to fetch data fromtable:

```
SELECTfield1,field2,...fieldNtable_name1,table_name2...
[WHEREcondition1[AND [OR]]condition2.....
```

- YoucanuseoneormoretablesseparatedbycommatoincludevariousconditionsusingaWHERE clause,butWHERE clauseisanoptionalpartofSELECTcommand.
- YoucanspecifyanyconditionusingWHEREclause.
- Youcanspecifymorethanoneconditionsusing**AND**or**OR**operators.
- A WHERE clause can be used along with DELETE or UPDATE SQL command also tospecifyacondition.

The **WHERE** clause works like an if condition in any programming language. This clause isusedtocomparegivenvaluewiththefieldvalueavailableintable.Ifgivenvaluefromoutsideisequal to theavailablefield valueintable, thenitreturns that row.

Hereisthelistofoperators,whichcanbeusedwith **WHERE**clause.AssumefieldAholds10 and field Bholds 20,then:

Operator	**Description**	**Example**
=	Checksifthevaluesoftwooperandsareequalornot,ifyesthenconditionbecomes true.	(A=B)isnottrue.
!=	Checksifthevaluesoftwooperandsareequal ornot,ifvaluesare notequalthenconditionbecomestrue.	(A!=B)istrue.
>	Checksifthevalueofleftoperandisgreaterthanthevalueofrightoperand,ifyes thencondition becomestrue.	(A>B)isnottrue.
<	Checksifthevalueofleftoperandislessthanthevalueofright operand,ifyesthenconditionbecomestrue.	(A< B)istrue.
>=	Checksifthevalueofleftoperandisgreaterthanorequaltothevalueof right operand,ifyesthenconditionbecomestrue.	(A>=B)isnottrue.
<=	Checksifthevalueofleftoperandislessthanorequaltothevalue ofrightoperand,ifyesthenconditionbecomestrue.	(A<= B)istrue.

TheWHEREclauseisveryusefulwhenyouwanttofetchselectedrowsfromatable,especiallywhenyou use**Join**.
Itisacommonpracticetosearchrecordsusing**PrimaryKey**tomakesearchfast.
Ifgivenconditiondoesnotmatchanyrecordinthetable,thenquerywouldnotreturnanyrow.

FetchingDatafromCommandPrompt:
Thiswilluse SQLSELECTcommandwithWHEREclause to fetch selected datafromtabletutorials_tbl.
Example:
Followingexamplewillreturnalltherecordsfrom**tutorials_tbl**tableforwhichauthornameis**Sanjay** :

```
>SELECT*fromtutorials_tblWHEREtutorial_author='Sanjay';
+-----------+-------------+---------------+---------------+
|tutorial_id|tutorial_title|tutorial_author|submission_date|
+-----------+-------------+---------------+---------------+
|        3|JAVATutorial|Sanjay           |2007-05-21     |
+-----------+-------------+---------------+---------------+
```

UnlessperformingaL**LIKE**comparisononastring,thecomparisonisnotcasesensitive.Youcanmakeyou rsearchcasesensitiveusing**BINARY**keywordas follows:

```
SELECT*fromtutorials_tblWHEREBINARYtutorial_author='sanjay';
```

LIKEClause

We have seen SQL **SELECT** command to fetch data fromtable. We can also use a conditionalclausecalled **WHERE**clauseto select required records.
A WHERE clause with equals sign (=) works fine where we want to do an exact match. Like if"tutorial_author = 'Sanjay'". But there may be a requirement where we want to filter out all theresultswheretutorial_authornameshouldcontain"jay".This can be handled usingSQL**LIKE**clausealongwithWHEREclause.
If SQL LIKE clause is used along with % characters, then it will work like a meta character (*)inUNIXwhilelistingoutallthefilesordirectoriesatcommand prompt.
Withouta%character,LIKEclauseisverysimilartoequalssignalongwithWHEREclause.
Syntax:
Here is generic SQL syntax of SELECT command along with LIKE clause to fetch data fromtable:

```
SELECT field1, field2,...fieldN table_name1,
table_name2...WHEREfield1LIKEcondition1[AND[OR]]
```

- □ YoucanspecifyanyconditionusingWHEREclause.
- □ YoucanuseLIKEclausealongwithWHEREclause.
- □ YoucanuseLIKEclauseinplaceofequalssign.
- □ WhenLIKEisusedalongwith%signthenitwillworklikeametacharactersearch.
- □ Youcanspecifymorethanoneconditionsusing**AND**or**OR**operators.
- □ AWHERE...LIKEclausecanbeused along withDELETEorUPDATESQLcommandalsoto specifyacondition.

UsingLIKEclauseatCommandPrompt:
This will use SQL SELECT command with WHERE...LIKE clause to fetch selected data fromtabletutorials_tbl.
Example:
Followingexamplewillreturnalltherecordsfrom**tutorials_tbl**tableforwhichauthornameendswith **jay**:

```
 SELECT*fromtutorials_tblWHEREtutorial_authorLIKE'%jay';
+------------+--------------+----------------+---------------+
|tutorial_id|tutorial_title|tutorial_author|submission_date|
+------------+--------------+----------------+---------------+
|          3|JAVATutorial|Sanjay              |2007-05-21      |
+------------+--------------+----------------+---------------+
```

GROUPBYClause

You can use **GROUP BY** to group values from a column, and, if you wish, perform calculationsonthatcolumn.You canuseCOUNT,SUM,AVG,etc.,functionsonthegroupedcolumn.
Tounderstand **GROUPBY** clause,consideran **employee_tbl** table,whichishavingthefollowingrecords:

```
 >SELECT*FROMemployee_tbl;
+------+------+-----------+------------------+
|id|name|work_date|daily_typing_pages|
+------+------+-----------+------------------+
|    1|John |2007-01-24 |                250|
|    2|Ram  |2007-05-27|                 220|
|    3|Jack|2007-05-06|                170 |
|    3|Jack|2007-04-06|                100 |
|    4|Jill|2007-04-06 |               220|
|    5|Zara|2007-06-06|                300 |
|    5|Zara|2007-02-06|                350 |
+------+------+-----------+
-------+7rows in set (0.00sec)
```

Now,supposebasedontheabovetablewewanttocountnumberofdayseachemployeedidwork.
Ifwe will writea SQLqueryasfollows,thenwewillgetthefollowingresult:

```
 SELECTCOUNT(*)FROMemployee_tbl;
+--------------------------+
|COUNT(*)                  |
+--------------------------+
|7                     |
```

Butthisisnotservingourpurpose,wewanttodisplaytotalnumberofpagestypedbyeachpersonseparately.Thisisdonebyusingaggregatefunctionsinconjunctionwitha**GROUPBY**clauseas follows:

```
SELECTname,COUNT(*)FROMemployee_tblGROUPBYname;

+++
|name|COUNT(*)|
+++
|Jack |2 |
|Jill |1 |
|John |1 |
|Ram|1 |
|Zara|2|
+++
5rowsinset (0.04 sec)
```

Wewillseemorefunctionalityrelatedto GROUPBYinotherfunctions likeSUM,AVG,etc.

COUNTFunction

COUNTFunctionisthesimplestfunctionandveryusefulincountingthenumberofrecords,whicharee expectedto bereturnedbyaSELECTstatement.
Tounderstand**COUNT**function,consideran**employee_tbl**table,whichishavingthefollowingrecords:

```
 >SELECT*FROMemployee_tbl;
+.........+.........+.................+..............................+
|id|name|work_date|daily_typing_pages|
+.........+.........+.................+..............................+
|    1|John |2007-01-24 |                250|
|    2|Ram  |2007-05-27|                 220|
|    3|Jack|2007-05-06|               170 |
|    3|Jack|2007-04-06|               100 |
|    4|Jill|2007-04-06 |              220|
|    5|Zara|2007-06-06|               300 |
|    5|Zara|2007-02-06|               350 |
+.........+.........+.................+
..........+7rows in set (0.00sec)
```

Now,supposebasedontheabovetableyouwanttocounttotalnumberofrowsinthistable,thenyoucan do itas follows:

```
 >SELECTCOUNT(*)FROMemployee_tbl;
+...............+
|COUNT(*)|
+...............+
|       7 |
+...............+
1rowinset (0.01sec)
```

Similarly,ifyouwanttocountthenumberofrecordsforZara,thenitcanbedoneasfollows:

```
SELECTCOUNT(*)FROMemployee_tblWHEREname="Zara";
+...............+
|COUNT(*)|
+----------+
|      2 |
+----------+
1rowinset (0.04sec)
```

NOTE: All the SQL queries are case insensitive so it does not make any difference if you giveZARAor ZarainWHEREcondition.

MAXFunction

MAXfunctionisusedtofindout therecordwithmaximumvalueamongarecordset. Tounderstand**MAX**function,consideran**employee_tbl**table,whichishavingthefollowingrecords :

```
 >SELECT*FROMemployee_tbl;
+------+------+------------+--------------------+
|id|name|work_date|daily_typing_pages|
+------+------+------------+--------------------+
|   1|John |2007-01-24 |              250|
|   2|Ram  |2007-05-27|               220|
|   3|Jack|2007-05-06|              170 |
|   3|Jack|2007-04-06|              100 |
|   4|Jill|2007-04-06 |             220|
|   5|Zara|2007-06-06|              300 |
|   5|Zara|2007-02-06|              350 |
+------+------+------------+
-------+7rows in set (0.00sec)
```

Now, suppose based on the above table you want to fetch maximum value ofdaily_typing_pages,thenyoucan dososimplyusingthefollowingcommand:

```
SELECTMAX(daily_typing_pages)FROMemployee_tbl;
+-------------------------+
|MAX(daily_typing_pages)|
+-------------------------+
|                 350 |
+
-------------------------
```

Youcanfindall the recordswithmaximum valueforeachname using**GROUPBY**clauseasfollows:

```
 SELECTid,name,MAX(daily_typing_pages)FROMemployee_tblGROUPBYname;
+------+------+-------------------------+
|id|name|MAX(daily_typing_pages)|
+------+------+-------------------------+
|   3|Jack |                  170 |
|   4|Jill|                  220 |
|1|John|                    250 |
|2|Ram|                     220 |
|5|Zara|                    350 |
++++5rows inset(0.00sec)
```

YoucanuseMINFunctionalongwithMAXfunctiontofindoutminimumvalueaswell.Tryout thefollowingexample:

```
SELECTMIN(daily_typing_pages)least,MAX(daily_typing_pages)maxFROMemployee_tbl;
+........+........+
|least|max  |
+........+........+
|100 |350 |
+........+........+
1rowinset (0.01sec)
```

MINFunction

MINfunctionisusedtofind out therecordwithminimum valueamongarecordset. Tounderstand**MIN**function,consideran**employee_tbl**table,whichishavingthefollowingrecord s:

```
 SELECT*FROMemployee_tbl;
+........+........+................+...........................+
|id|name|work_date|daily_typing_pages|
+........+........+................+...........................+
|    1|John |2007-01-24 |                250|
|    2|Ram  |2007-05-27|                 220|
|    3|Jack|2007-05-06|                170 |
|    3|Jack|2007-04-06|                100 |
|    4|Jill|2007-04-06 |               220|
|    5|Zara|2007-06-06|                300 |
|    5|Zara|2007-02-06|                350 |
+........+........+................+
..........+7rows in set (0.00sec)
```

Now,supposebasedontheabovetableyouwanttofetchminimumvalueofdaily_typing_pages,thenyou cando sosimplyusingthefollowingcommand:

```
SELECTMIN(daily_typing_pages)FROMemployee_tbl;
+...................................+
|MIN(daily_typing_pages)|
+...................................+
|                    100 |
+
...................................
```

You can find all the records with minimum value for each name using **GROUP BY** clause asfollows:

```
SELECTid,name,MIN(daily_typing_pages)FROMemployee_tblGROUPBYname;
+.......+.......+.........................+
|id|name|MIN(daily_typing_pages)|
+.......+.......+.........................+
|   3|Jack |                  100 |
|   4|Jill|                  220 |
|   1|John|                   250 |
|   2|Ram|                    220 |
|   5|Zara|                  300 |
+.......+.......+
.......+5rows inset(0.00sec)
```

Youcanuse**MIN**Functionalongwith**MAX**functiontofindoutminimumvalueaswell.Tryout thefollowingexample:

```
SELECTMIN(daily_typing_pages)least,MAX(daily_typing_pages)maxFROMemployee_tbl;
+.......+......+
|least|max  |
+.......+......+
|100 |350 |
+.......+......+
1rowinset (0.01sec)
```

AVGFunction

AVGfunctionisusedtofindouttheaverageofafieldinvariousrecords.

Tounderstand**AVG**function, consideran**employee_tbl** table,whichishavingfollowingrecords:

```
 SELECT*FROMemployee_tbl;
+.......+.......+............+.....................+
|id|name|work_date|daily_typing_pages|
+.......+.......+............+.....................+
|   1|John |2007-01-24 |              250|
|   2|Ram  |2007-05-27|               220|
|   3|Jack|2007-05-06|               170 |
|   3|Jack|2007-04-06|               100 |
|   4|Jill|2007-04-06 |             220|
|   5|Zara|2007-06-06|               300 |
|   5|Zara|2007-02-06|               350 |
+.......+.......+............+
.......+7rows in set (0.00sec)
```

Now,supposebasedontheabovetableyouwanttocalculateaverageofallthedialy_typing _pages,thenyoucan dosobyusingthefollowingcommand:

```
 SELECTAVG(daily_typing_pages)FROMemployee_tbl;
+--------------------------+
|AVG(daily_typing_pages)|
+--------------------------+
|             230.0000|
+
--------------------------
```

You can take average of various records set using **GROUP BY** clause. Following example willtakeaveragealltherecordsrelatedtoasinglepersonandyouwillhaveaveragetypedpagesbyevery person.

```
 SELECTname,AVG(daily_typing_pages)FROM employee_tblGROUPBYname;
+------+--------------------------+
|name|AVG(daily_typing_pages)|
+------+--------------------------+
|Jack |               135.0000 |
|Jill |              220.0000 |
|John |               250.0000 |
|Ram|                 220.0000 |
|Zara|                325.0000 |
+------+
-------+5rowsinset(0.20sec)
```

SUMFunction

SUMfunctionisusedtofindout thesumofafieldinvariousrecords.
TounderstandSUMfunction,consideran**employee_tbl**table,whichishavingthefollowingrecords:

```
SELECT*FROMemployee_tbl;
+------+------+-------------+--------------------+
|id|name|work_date|daily_typing_pages|
+------+------+-------------+--------------------+
|    1|John |2007-01-24 |                 250|
|    2|Ram  |2007-05-27|                  220|
|    3|Jack|2007-05-06|                 170 |
|    3|Jack|2007-04-06|                 100 |
|    4|Jill|2007-04-06 |                220|
|    5|Zara|2007-06-06|                 300 |
|    5|Zara|2007-02-06|                 350 |
+------+------+-------------+
-------+7rows in set (0.00sec)
```

Now,supposebasedontheabovetableyouwanttocalculatetotalofallthedialy_typing_pages,thenyou cando so byusingthefollowingcommand:

```
SELECTSUM(daily_typing_pages)FROMemployee_tbl;
+--------------------------+
|SUM(daily_typing_pages)|
+-----------------------------+
|                1610 |
+
-----------------------------
```

You can take sum of various records set using **GROUP BY** clause. Following example will sumupalltherecordsrelatedtoasinglepersonandyouwillhavetotaltypedpagesbyeveryperson.

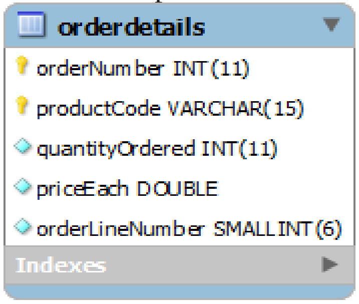
```
SELECTname,SUM(daily_typing_pages)FROMemployee_tblGROUPBYname;
+------+---------------------------+
|name|SUM(daily_typing_pages)|
+------+---------------------------+
|Jack |                  270 |
|Jill |                 220 |
|John |                  250 |
|Ram|                    220 |
|Zara|                  650|
+------+
--------+5rowsinset(0.17sec)
```

HAVINGclause

TheHAVING clause is used in the SELECT statement to specify filter conditions for group ofrows or aggregates. TheHAVING clause is often used with the GROUP BY clause. WhenusingwiththeGROUPBYclause,youcanapplyafilterconditiontothecolumnsthatappearinthe GROUP BY clause. If the GROUP BY clause is omitted, theHAVING clause behaves likethe WHERE clause. Notice that theHAVING clause applies the condition to each group ofrows,whiletheWHEREclauseappliestheconditiontoeachindividual row.

ExamplesofusingHAVINGclause

Let‟stake alookatanexampleof using HAVINGclause. Wewillusetheorderdetailstableinthesampledatabaseforthesakeofdemonstration.



orderdetails

orderNumber INT(11)

productCode VARCHAR(15)

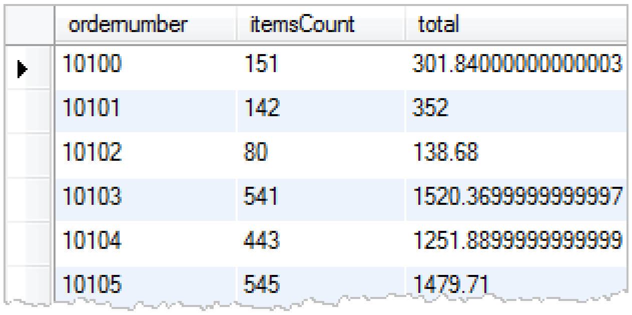
quantityOrdered INT(11)

priceEach DOUBLE

orderLineNumber SMALLINT(6)

We can use theGROUP BY clause to get order number, the number of items sold per order andtotalsalesforeach:

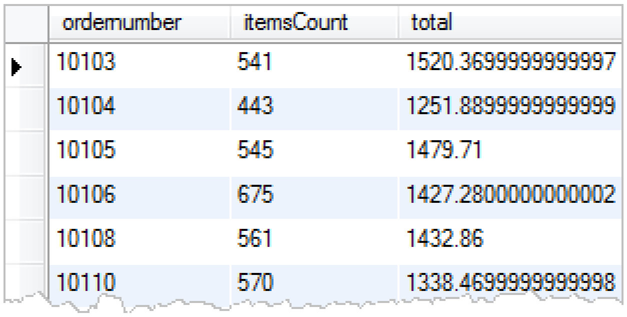
```
SELECT
      ordernumber,SUM(quantityOrdered)A
      S itemsCount,SUM(priceeach)AS total
FROM
orderdetailsGROUPBYor
```



ordernumber	itemsCount	total
10100	151	301.84000000000003
10101	142	352
10102	80	138.68
10103	541	1520.3699999999997
10104	443	1251.8899999999999
10105	545	1479.71

Now,wecanfindwhichorderhastotalsalesgreaterthan$1000.WeusetheHAVINGclauseontheaggregateas follows:

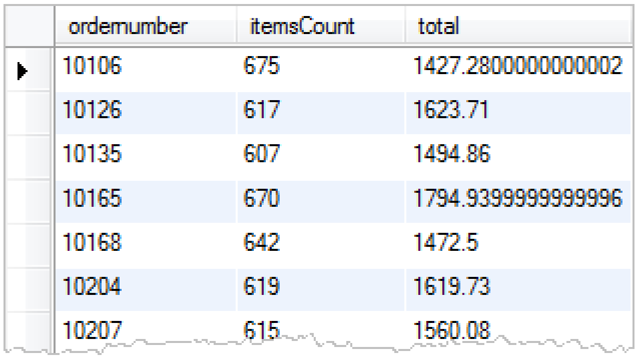
```
SELECT
      ordernumber,SUM(quantityOrdered)A
      S itemsCount,SUM(priceeach)AS total
       FROM
      orderdetailsGROUPBYor
      dernumber
```



ordernumber	itemsCount	total
10103	541	1520.3699999999997
10104	443	1251.8899999999999
10105	545	1479.71
10106	675	1427.2800000000002
10108	561	1432.86
10110	570	1338.4699999999998

We can construct a complex condition in theHAVING clause using logical operators suchas OR and AND. Suppose we want to find which order has total sales greater than $1000 andcontainsmorethan600 items,wecan usethefollowingquery:

```
SELECT
      ordernumber,sum(quantityOrdered)A
      S itemsCount,sum(priceeach)AS total
       FROM
       orderdetailsGROUPBYor
       dernumber
```



ordernumber	itemsCount	total
10106	675	1427.2800000000002
10126	617	1623.71
10135	607	1494.86
10165	670	1794.9399999999996
10168	642	1472.5
10204	619	1619.73
10207	615	1560.08

TheHAVING clause is only useful when we use it with the GROUP BY clause to generate theoutput of the high-level reports. For example, we can use theHAVING clause to answer somekindsofquerieslikegivemeall theordersinthismonth,thisquarterandthisyearthathavetotalsalesgreater than 10K.

UPDATEQuery

There may be a requirement where existing data in atable needs to be modified. You can do sobyusingSQL**UPDATE**command.Thiswillmodifyanyfield valueof anytable.
Syntax:
Here isgenericSQLsyntaxofUPDATE commandtomodifydata intotable:

```
UPDATEtable_nameSETfield1=new-value1,field2=new-
value2[WHEREClause]
```

□ Youcanupdateoneormorefieldaltogether.
□ YoucanspecifyanyconditionusingWHEREclause.
□ Youcanupdatevaluesinasingletableatatime.

TheWHEREclauseisveryusefulwhenyouwanttoupdateselectedrowsinatable.Updating DatafromCommand Prompt:
This will use SQL UPDATE command with WHERE clause to update selected data intotabletutorials_tbl.
Example:
Followingexamplewillupdate**tutorial_title**fieldforarecordhavingtutorial_idas3.

```
UPDATEtutorials_tbl
  SETtutorial_title='LearningJAVA'
    WHEREtutorial_id=3;
```

DELETEQuery

If you want to delete a record from anytable, then you can use SQL command **DELETEFROM**. You can use this command at > prompt as well as in any script like PHP.**Syntax:**
Here isgenericSQLsyntaxof DELETEcommand to deletedatafromatable:

```
DELETEFROMtable_name[WHEREClause]
```

□ IfWHEREclauseisnotspecified,thenalltherecordswillbedeletedfromthegiventable.
□ YoucanspecifyanyconditionusingWHEREclause.
□ Youcandeleterecordsinasingletableatatime.

TheWHEREclauseisveryusefulwhenyouwanttodeleteselectedrowsinatable.Deleting DatafromCommand Prompt:
This will use SQL DELETE command with WHERE clause to delete selected data intotabletutorials_tbl.
Example:
Followingexamplewill deletearecordintotutorial_tblwhosetutorial_idis3.

```
DELETEFROMtutorials_tblWHEREtutorial_id=3;
```

Createtable**location**(location_idnumeric(3)primarykey,regional_groupvarchar(15));

Createtable**department**(Department_IDnumeric(2)primarykey,namevarchar(20),location_idint, foreignkey(location_id) referenceslocation(location_id));

Createtable**job**(job_IDnumeric(3)primarykey,functionvarchar(20));

Createtable**employee**(employee_IDnumeric(4)primarykey,last_namevarchar(20),first_namevarchar(20),middle_name varchar(20),job_id numeric(3),manager_id varchar(20), hired_datedate,salarynumeric(6),commnumeric(4),department_idnumeric(2)notnull,FOREIGN KEY(job_id) REFERENCES job(job_id),FOREIGN KEY (department_id) REFERENCESdepartment(department_id));

1. Listthedetailsabout“SMITH”

   Select*fromemployeewherelast_name=‘SMITH’;

2. Listouttheemployeeswhoareworkingindepartment20

   Select*fromemployeewheredepartment_id=20

3. Listouttheemployeeswhoareearningsalarybetween3000and4500

   Select*fromemployeewheresalarybetween3000and4500

4. Listouttheemployeeswhoareworkingindepartment10or20

   Select*fromemployeewheredepartment_idin(10,20)

5. Findouttheemployeeswhoarenotworkingindepartment10or30

   Selectlast_name,salary,comm,department_idfromemployeewheredepartment_idnotin (10,30)

6. Listouttheemployeeswhosenamestartswith“S”

   Select*fromemployeewherelast_namelike'S%';

7. Listouttheemployeeswhosenamestartwith“S”andendwith“H”

   Select*fromemployeewherelast_nameLike'S%H';

8. Listouttheemployeeswhosenamelengthis5andstartwith“S”

   Select*fromemployeewherelast_namelike 'S___';

9. Listouttheemployeeswhoareworkingindepartment10anddrawthesalariesmorethan 3500

   Select*fromemployeewheredepartment_id=10andsalary>3500

10. Listouttheemployeeswhoarenotreceivingcommission.

    Select*fromemployeewherecommissionisNull

11. Listouttheemployeeid,lastnameinascendingorderbasedontheemployeeid.

    Selectemployee_id,last_namefromemployeeorderbyemployee_id

12. Listouttheemployeeid,nameindescendingorderbasedonsalarycolumn

    Selectemployee_id,last_name,salaryfromemployeeorderbysalarydesc

13. Listouttheemployeedetailsaccordingtotheirlast_nameinascendingorderandsalaries in descendingorder

Conclusion: Thuswehave studied touse &implementvariousDMLqueries.

FAQ:

1. ExplainDML.
2. ExplainINSERTcommandwithsyntax.
3. ExplainDELETEcommandwithsyntax.
4. ExplainUPDATEcommandwithsyntax.
5. ExplainSELECTcommandwithsyntax.
6. Enlistdifferentcomparisonsoperator.Explainwithexample.
7. EnlistdifferentLogicaloperator.Explainwithexample.
8. ExplainOrderbyclause.
9. EnlistdifferentAggregation function.Explain withexample.

AssignmentNo.	**3**
Title	**SQLQueriesalltypesofJoin,Sub-QueryandView:** Writeatleast10SQLqueriesforsuitabledatabaseapplicationusingSQLDMLstatements.
PROBLEM STATEMENT/DEFINITION	Writeatleast10SQLqueriesforsuitabledatabaseapplicationusingSQLDMLstatements
Objectives	To understand • Types of joins. • Subquery and its types. • Complex views
Software packages and hardware apparatus used	MySQL/Oracle PC with the configuration as Latest Version of 64 bit Operating Systems, Open Source Fedora-GHz. 8 G.B. RAM, 500 G.B. HDD, 15"Color Monitor, Keyboard, Mouse
References	Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", 5th Edition, McGraw Hill Publishers, 2002, ISBN 0-07-120413-X Connally T., Begg C., "Database Systems", 3rd Edition, Pearson Education, 2002, ISBN 81-7808-861-4 mysql.com/docs/mysql-tutorial-excerpt-5.1-en.pdf
STEPS	Refer to details
Instructions for writing journal	• Date • Title • Problem Definition • Learning Objective • Learning Outcome • Theory-Related concept,Architecture,Syntax etc • Class Diagram/ER diagram • Test cases • Program Listing • Output • Conclusion

AssignmentNo:3

Title:-Designat least10SQLqueriesforsuitabledatabaseapplication usingSQLDMLstatements: alltypesofJoin,Sub-QueryandView.

Objectives:-a)Types of joins. b) Subquery and its types c)Complex views

THEORY:SQL‒Join

Theabilityofrelational„join‟operatorisanimportantfeatureofrelationalsystems.Ajoin makesitpossibletoselect datafrommorethantablebymeansofasinglestatement.Thisjoiningoftables maybedoneinamanyways.

TypesofJOIN

1) Inner
2) Outer(left,right,full)
3) Cross

1) Innerjoin :

- Alsoknownasequijoin.
- Statementsgenerallycomparestwocolumnsfromtwocolumnswiththeequivalenceoperator=.
- Thistypeofjoincanbeusedinsituationswhereselectingonlythoserowsthathavevaluesin common in thecolumnsspecifiedin theONclause, isrequired.

• Syntax:

(ANSIstyle)

SELECT<columnname1>,<columnname2><columnNameN>FROM<tablename1>INNER JOIN <tablename2>ON <tablename1>.<columnname> = <tablename2>.<columnname>WHERE<condition>ORDERBY<columnname1>;

(thetastyle)

SELECT<columnname1>,<columnname2><columnNameN>FROM<tablename1>, <tablename2>WHERE<tablename1>.<columnname>=<tablename2>.<columnname>AND<condition>ORDERBY<columnname1>;

- Listtheemployeedetailsalongwithbranchnamestowhichtheybelong.

Emp(empno,fname,lname,dept,desig,branchno)Branch(bname,branchno)

Selecte.empno,e.fname,e.lname,e.dept,b.bname,e.desigfromempeinnerjoinbranchbonb.branchno=e.branchno;

Selecte.empno,e.fname,e.lname,e.dept,b.bname,e.desigfromempe,branchbonwhereb.branchno=e.branchno;

Eg.Listthecustomersalongwiththeaccountdetailsassociatedwiththem.Customer(cust no,fname,lname)

Acc_cust_dtls(fdno,custno)

Acc_mstr(accno,branchno,curbal)Bran

ch_mstr(name,branchno)

- Selectc.custno,c.fname,c.lname,a.accno,a.curbal,b.branchno,b.namefromcustomercinnerjoin acc_cust_dtls k on c.custno=k.custno inner join acc_mstr a on k.fdno=a.accno inner joinbranch bonb.branchno=a.branchnowhere c.custno like„C%‟orderbyc.custno;
- Selectc.custno,c.fname,c.lname,a.accno,a.curbal,b.branchno,b.namefromcustomerc,acc_cust_dtlsk,acc_mstra,branchbwherec.custno=k.custnoandk.fdno=a.accnoandb.branchno=a.branchnoand c.custnolike„C%‟orderbyc.custno;

OuterJoin

Outer joins are similar to inner joins, but give a little bit more flexibility when selecting data fromrelatedtables.Thistypeofjoinscanbeusedinsituationswhereitisdesired,toselect allrowsfromthetable on left(or right, or both) regardless of whether the other table has values in common & (usually)enterNULLwheredatais missing.

- Tables

Emp_mstr(empno,fname,lname,dept)C

ntc_dtls(codeno,cntc_type,cntc_data)

LeftOuterJoin
Listtheemployeedetailsalongwiththecontactdetails(ifany)usingleftouterjoin.

- Selecte.empno,e.fname,e.lname,e.dept,c.cntc_type,c.cntc_datafromemp_mstreleftjoincntc_dtlscon e.empno=c.codeno;
- Selecte.empno,e.fname,e.lname,e.dept,c.cntc_type,c.cntc_datafromemp_mstrecntc_dtlscwhere e.empno=c.codeno(+);

Alltheemployeedetailshavetobelistedeventhoughtheircorrespondingcontactinformationisnotpresent. This indicates all the rows from the first table will be displayed even though there exists nomatchingrows in thesecond table.

Rightouterjoin

Listtheemployeedetailswithcontactdetails(ifanyusingrightouterjoin.

- TablesEmp_mstr(empno,fnam

e,lname,dept)Cntc_dtls(codeno,cntc_t

ype,cntc_data)

- Selecte.empno,e.fname,e.lname,e.dept,c.cntc_type,c.cntc_datafromemp_mstrerightjoincntc_dtlscon e.empno=c.codeno;

- Selecte.empno,e.fname,e.lname,e.dept,c.cntc_type,c.cntc_datafromemp_mstrecntc_dtlscwhere e.empno(+)=c.codeno;

SincetheRIGHTJOINreturnsalltherowsfromthesecondtableeveniftherearenomatchesinthefirsttable.

Crossjoin

A cross join returns what known as a Cartesian Product. This means that the join combineseveryrow fromthelefttablewitheveryrowintheright table.Ascanbeimagined,sometimes

join can be used in situation where it is desired, to select all possible combinations of rows &columnsfrom bothtables.Thekindofjoinisusuallynot preferredasitmayrunforaverylongtime&produceahugeresultsetthatmaynotbeuseful.

- Createareportusingcrossjointhatwilldisplaythematurityamountsforpredefineddeposits, basedonmin&maxperiodfixed/timedeposit.
- TablesTem_fd_amt(fd_a

mt)Fd_mstr(minprd,maxprd,intr

ate)

- Selectfd_amt,s.minprd,s.maxprd,s.intrate,round(t.fd_amt+(s.intrate/100)*(s.minprd/365)))"amount_min_period",round(t.fd_amt+(s.intrate/100)*(s.maxprd/365)))"amount_max_period"from fd_mstrs crossjointem_fd_amtt;
- Select t.fd_amt, s.minprd, s.maxprd, s.intrate, round(t.fd_amt+(s.intrate/100) * (s.minprd/365)))"amount_min_period",round(t.fd_amt+(s.intrate/100)*(s.maxprd/365)))"amount_max_period"

  fromfd_mstrs,tem_fd_amtt;

Self join

- Insomesituation,itisnecessarytojointoitself,asthoughjoining2separatetables.
- ThisisreferredtoasselfjoinExample
- Emp_mgr(empno,fname,lname,mgrno)

- Selecte.empno,e.fname,e.lname,m.fname"manager"from

  emp_mgre,emp_mgrmwheree.mgrno=m.empno;

:

Employee(Eno,Ename,Deptno,Salary)

Eno=pk,Deptno=fkDepartment(Deptno,Dname) Deptno=pk

Implementalljoinoperation–crossjoin,naturaljoin,equi join,left outer,rightouterjoinetc&Write SQLQueries for followingquestions

i) Listofemployeenamesof'Computer'department.

ii) FindtheEmployeewho"sSalaryabove50000ofeachdepartment.

iii) Finddepartmentnameofemployeename'Amit'.

Conclusion: Thuswehavestudiedtouse&implement variousjoinoperationwithnestedqueries.

FAQ:

1. ExplainJoinFunction.
2. Enlistthedifferenttypesofjoinoperations.
3. Explain CROSSJoin explainwithexample.
4. ExplainNaturaljoinexplainwithexample.
5. ExplainInnerjoinexplainwithexample.
6. ExplainOuterjoinexplainwithexample.
7. WhatistheuseofnestedQuery.ExplainwithExample.

AssignmentNo.	**4**
Title	Unnamed PL/SQL code block: Use of Control structure andExceptionhandlingismandatory.
PROBLEM STATEMENT/DEFINITION	WriteaPL/SQLcodeblocktocalculatetheareaofacircleforavalueofradiusvaryingfrom5to 9. Store the radius and the corresponding values of calculated area in an empty table namedareas,consistingoftwocolumns,radiusandarea.
Objectives	• Understand the control structure • Understand exception handling in PL/SQL
Software packages and hardware apparatus used	MySQL/Oracle PC with the configuration as Latest Version of 64 bit Operating Systems, Open Source Fedora-GHz. 8 G.B. RAM, 500 G.B. HDD, 15"Color Monitor, Keyboard, Mouse
References	Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", 5th Edition, McGraw Hill Publishers, 2002, ISBN 0-07-120413-X Connally T., Begg C., "Database Systems", 3rd Edition, Pearson Education, 2002, ISBN 81-7808-861-4 mysql.com/docs/mysql-tutorial-excerpt-5.1-en.pdf
STEPS	Refer to details
Instructions for writing journal	• Date • Title • Problem Definition • Learning Objective • Learning Outcome • Theory-Related concept,Architecture,Syntax etc • Class Diagram/ER diagram • Test cases • Program Listing • Output • Conclusion

AssignmentNo:4

Title:-Unnamed PL/SQL code block: Use of Control structure and Exception handling is

mandatory.Write a PL/SQLblockofcode forthefollowingrequirements:-
Schema:
1. Borrower(Rollin,Name,DateofIssue,NameofBook,Status)
2. Fine(Roll_no,Date,Amt)

- Acceptroll_no&nameofbookfromuser.
- Checkthenumberofdays(fromdateofissue),ifdaysarebetween15to30thenfine amount will be Rs5per day.
- Ifno.ofdays>30,perdayfinewillbeRs50perday&fordayslessthan30,Rs.5perday.
- AftersubmittingthebookstatuswillchangefromItoR.
- Ifconditionoffineistrue,thendetailswillbestoredintofinetable.

FrametheproblemstatementforwritingPL/SQLblockinlinewithabovestatement.Objective:-

a)Understand the control structure b)Understand exception handling in PL/SQL

Theory:
Introduction:-PL/SQL

Thedevelopmentofdatabaseapplicationstypicallyrequireslanguageconstructssimilartothosethatcanbefoundinprogramminglanguagessuchas C,C++,orPascal.Theseconstructsarenecessaryinorder to implement complex data structures and algorithms. A major restriction of the databaselanguage SQL, however, is that many tasks cannot be accomplished by using only the providedlanguageelements.

PL/SQL(Procedural Language/SQL)isa proceduralextension ofOracle-SQLthatoffers languageconstructssimilar to thosein imperativeprogramminglanguages.

Or

A PL/SQLis aprocedural language extension to the SQL in which you can declare and use thevariables,constants,doexceptionhandlingandyoucanalsowritetheprogrammodulesintheformofPL/SQL subprograms.PL/SQL combines the features of a procedural language with structured querylanguage

PL/SQL allows users and designers to develop complex database applications that require the usage ofcontrol structuresandproceduralelementssuchasprocedures,functions,andmodules.

The basic construct in PL/SQL is a block. Blocks allow designers to combine logically related (SQL-)statementsintounits.Inablock,constantsandvariablescanbedeclared,andvariablescanbeusedtostore query results. Statements in a PL/SQL block include SQL statements, control structures (loops),conditionstatements(if-then-else),exception handling,andcalls ofother PL/SQLblocks.

PL/SQLblocksthatspecifyproceduresandfunctionscanbegroupedintopackages.Apackageissimilartoamoduleandhasaninterfaceandanimplementationpart.Oracleoffersseveralpredefinedpackages,forexample,input/outputroutines,filehandling,jobschedulingetc.(seedirectory
$ORACLEHOME/rdbms/admin).

Anotherimportantfeatureof PL/SQListhat itoffersamechanismtoprocess queryresults in atuple-oriented way, that is, one tuple at a time. For this, cursors are used. A cursor basically is a pointer to aqueryresultandisusedto readattribute valuesof selected tuples into variables.Acursor typicallyis usedincombinationwithaloopconstructsuchthateachtuplereadbythecursorcanbeprocessedindividually.

Insummary,themajorgoals of PL/SQLareto

- IncreasetheexpressivenessofSQL,
- Processqueryresultsinatuple-orientedway,
- OptimizecombinedSQLstatements,
- Developmodulardatabaseapplicationprograms,
- Reuseprogramcode,and
- Reducethecostformaintainingandchangingapplications

AdvantagesofPL/SQL:-

Followingaresomeadvantagesof Pl/SQL

1) SupportforSQL:-PL/SQListheprocedurallanguageextensiontoSQLsupportsallthefunctionalitiesof SQL.

2) Improved performance:- In SQL every statement individually goes to the ORACLE server, getprocessed and then execute. But in PL/SQL an entire block of statements can be sent to ORACLEserver at one time, where SQL statements are processed one at atime.PL/SQL block statementsdrasticallyreducecommunicationbetweentheapplicationandORACLE.Thishelpsinimprovingtheperformance.

3) Higher Productivity:- Users use procedural features to build applications.PL/SQL code is written inthe form of PL/SQL block.PL/SQL blocks can also used in other ORACLE Forms, ORACLE reports.Thiscodereusabilityincreasestheprogrammersproductivity.

4) Portability:-Applicationswritten inPL/SQLare portable.WecanportthemfromoneenvironmenttoanycomputerhardwareandoperatingsystemenvironmentrunningORACLE.

5) IntegrationwithORACLE:-BothPL/SQLandORACLE are SQLbased.PL/SQLvariables havedatatypesnativetotheoracleRDBMSdictionary.ThisgivestightintegrationwithORACLE.

FeaturesofPL/SQL:-

1) WecandefineandusevariablesandconstantsinPL/SQL.

2) PL/SQLprovidescontrolstructures to controltheflowofaprogram.ThecontrolstructuressupportedbyPL/SQLare if..Then,loop, for..loop and others.

3) We candorowbyrowprocessingofdatainPL/SQL.PL/SQLsupports rowbyrowprocessingusingthemechanismcalled cursor.

4) Wecanhandlepre-definedanduser-definederrorsituations.Errorsarewarningsandcalledasexceptionsin PL/SQL.

5) Wecanwritemodularapplicationbyusingsubprograms.

Thestructure ofPL/SQLprogram:-

The basic unit ofcodeinanyPL/SQLprogramis ablock.AllPL/SQLprograms arecomposedofblocks.Theseblockscan bewritten sequentially.

Thestructure ofPL/SQLblock:-

```
DECLARE

        Declaration
sectionBEGIN
        Executable
sectionEXCEPTION
        Exception handlingsection
END;
```

Where

1) Declarationsection
   PL/SQLvariables,types,cursors,andlocalsubprograms aredefined here.
2) Executablesection
   Procedural and SQL statements are written here. This is the main section of the block.Thissection is required.
3) Exceptionhandlingsection
   Errorhandlingcodeiswrittenhere
   Thissectionisoptionalwhetheritisdefinedwithinbodyoroutsidebodyofprogram.

ConditionalstatementsandLoopsusedin PL/SQL

Conditionalstatementscheckthevalidityofaconditionandaccordinglyexecuteasetofstatements.Theconditionalstatementssupported byPl/SQLis

1) IF..THEN
2) IF..THEN..ELSE
3) IF..THEN..ELSIF

1) IF..THEN

```
Syntax1:-
        IfconditionTHEN
        StatementlistEND
        IF;
```

2) IF..THEN..ELSE

```
Syntax2:-
        IFconditionTHEN
                Statementlist
        ELSE
                Statements
        ENDIF;
```

3) IF..THEN..ELSIF

```
Syntax3:-
        IfconditionTHEN
                Statement
        listELSIFconditionTH
        EN
                Statementlist
        ELSE
                Statementlist
        END
```

```
IF;END

IF;
```

2) CASEExpression:CASEexpressioncanalsobeusedtocontrolthebranchinglogicwithinPL/ SQLblocks.Thegeneralsyntax is

```
CASE
WHEN<expression>THEN<statements>;
WHEN<expression>THEN<statements>;
.
.ELS
E
<statements>;
ENDCASE;
```

HereexpressioninWHENclauseisevaluatedsequentially.WhenresultofexpressionisTRUE,thencorrespondingseto fstatementsareexecutedandprogram flowgoes toENDCASE.

ITERATIVEConstructs: Iterativeconstructsareusedtoexecuteasetofstatementsrespectively.Theiterativeconstr uctssupported byPL/SQLarefollows:

1) SIMPLELOOP
2) WHILELOOP
3) FORLOOP

1) TheSimpleLOOP:Itisthesimplestiterativeconstructandhassyntaxlike:LOOP

```
 Statements
ENDLOOP;
```

TheLOOPdoesnotfacilitateacheckingforaconditionandsoitisanendlessloop.Toendtheiterations,th eEXITstatement canbeused.

```
LOOP
        <statementlist>
                IFconditionTHEN
                EXIT;
                ENDIF;
ENDLOOP;
```

Thestatementshereisexecutablestatements,whichwillbeexecutedrepeatedlyuntiltheconditiongivenifIF..T HENevaluatesTRUE.

2) THEWHILELOOP
TheWHILE…LOOPisaconditiondrivenconstructi.etheconditionisapartoftheloopconstruct and nottobecheckedseparately.TheloopisexecutedaslongastheconditionevaluatestoTRUE.Thesynta

xis:-

```
WHILEconditionLOOP
     Statements
ENDLOOP;
```

Theconditionisevaluatedbeforeeachiterationofloop.IfitevaluatestoTRUE,sequenceofstatementsare executed. If the condition is evaluated to FALSE or NULL, the loop is finished and the controlresumesafter theENDLOOPstatement.

3) THE FOR LOOP :The number of iterations for LOOP and WHILE LOOP is not known in advance.THEnumberofiterationsdependsontheloopcondition.TheFORLOOPcanbeusedtohaveadefinitenumbersofiterations.

Thesyntaxis:-

```
ForloopcounterIN[REVERSE]Lowbound..HighboundLOOPState
ments;
End loop;
```

Where

- loopcounter¯istheimplicitlydeclaredindexvariableasBINARY_INTEGER.
- Lowboundandhighboundspecifythenumberofiteration.
- Statements:-Arethecontentsoftheloop

EXCEPTIONS:-Exceptionsare errorsorwarningsin aPL/SQLprogram.PL/SQLimplements errorhandlingusingexceptions and exception handler.
Exceptions are the run time error that a PL/SQL program may encounter.Therearetwotypesof exceptions

1) Predefinedexceptions
2) Userdefinedexceptions

1) Predefinedexceptions:-
PredefinedexceptionsaretheerrorconditionthataredefinedbyORACLE.Predefined exceptions cannot be changed. Predefined exceptions correspond to common SQL errors.The predefined exceptions are raised automatically whenever a PL/SQL program violates an ORACLErule.

2)UserdefinedExceptions:-Auser defined exceptions isan error orawarningthatis definedbytheprogram.Userdefinedexceptionscanbe definein thedeclaration section ofPL/SQLblock. User defined exceptions are declared in the declarative section of a PL/SQL block. Exceptions have atypeException and scope**.**

Syntax :

```
DECLARE
        <ExceptionName>EXCEPTION;
BEGIN
        ….
        RAISE<ExceptionName>
        …
EXCEPTION
        WHEN<Exceptionname>THEN
        <Action>
END;
```

Exception Handling

A PL/SQL block may contain statements that specify exception handling routines. Each error orwarning during the execution of a PL/SQL block raises an exception. One can distinguish between twotypesofexceptions:

- **Systemdefinedexceptions**
- **Userdefinedexceptions**(whichmustbedeclaredbytheuserinthedeclarationpartofablockwheretheexception isused/implemented)

System defined exceptions are always automatically raised whenever corresponding errors or warnings occur. User defined exceptions, in contrast, must be raised explicitly in a sequence of statements usingraise<exceptionname>.Afterthekeywordexceptionat theendofablock,userdefinedexception handlingroutinesareimplemented.An implementation has thepatternwhen<exceptionname>then<sequenceofstatements >;

The most common errors that can occur during the execution of PL/SQL programs are handled bysystemdefinedexceptions.Thetablebelowlistssomeoftheseexceptionswiththeirnamesandashort description.

Oracle Error	**EquivalentException**	**Description**
ORA-0001	DUP_VAL_ON_INDEX	Uniqueconstraintviolated.
ORA-0051	TIMEOUT_ON_RESOURSE	Time-outoccurredwhilewaitingforrecourse
ORA-0061	TRANSACTION_BACKED_OUT	Thetransactionwasrolledbackto duetodeadlock.
ORA-1001	INVALID_CURSOR	Illegalcursoroperation.
ORA-1012	NOT_LOGGED_ON	NotconnectedtoOracle.
ORA-1017	LOGIN_DENIED	Invalidusername/passward
ORA-1403	NO_DATA_FOUND	Nodatafound.
ORA-1410	SYS_INVALID_CURSOR	Conversiontoauniversalrowedfailed.
ORA-1422	TOO_MANY_ROWS	A SELECT…INTO statement matchesmorethanonerow.
ORA-1476	ZERO_DIVIDE	Divisionbyzero.
ORA-1722	INVALID_NUMBER	Conversiontoanumberfailed.
ORA-6500	STORAGE_ERROR	Internal PL/SQL error raised ifPL/SQLrunsoutof memory.
ORA-6501	PROGRAM_ERROR	InternalPL/SQLerror.
ORA-6502	VALUE_ERROR	Truncation,arithmeticorconversionerror.
ORA-6504	ROWTYPE_MISMATCH	HostcursorvariableandPL/SQLcursorvariablehaveincompatible rowtype
ORA-6511	CURSOR_ALREADY_OPEN	Attempttoopenacursorthatis alreadyopen.
ORA-6530	ACCESS_INTO_NULL	Attempttoassignvaluestotheattributes ofaNULLobject.
ORA-6531	COLLECTION_IS_NULL	Attempt to apply collection methodsotherthanEXISTStoaNULL PL/SQLtableor varray.
ORA-6532	SUBSCRIPT_OUTSIDE_LIMIT	Referencetoanestedtableorvarray indexoutsidethedeclaredrange.
ORA-6533	SUBSCRIPT_BEYOND_COUNT	Reference to a nested table or

		varrayindexhigherthanthenumberof

		elementsinthecollection
ORA-6592	CASE_NOT_FOUND	NomatchingWHENclauseinaCASEstatementisfound
ORA-30625	SELF_IS_NULL	AttempttocallamethodonaNULLobjectinstance

Syntax:-

```
        <Exception_name>Exception;
```

HandlingExceptions:-Exceptionshandlersforalltheexceptionsarewrittenintheexceptionhandlingsection ofaPL/SQLblock.

Syntax:-

```
        Exception
                When exception_name
                        thenSequence_of_statements1;
                When exception_name
                        thenSequence_of_statements2;
                When exception_name
                        thenSequence_of_statements3;
        End;
```

Example:

```
        Declare
                emp sal EMP.SAL
                %TYPE;empnoEMP.EMPNO
                %TYPE;
                too_high_salexception;
        begin
                select EMPNO, SAL into emp no, emp
                salfromEMPwhereENAME=‟KING‟;
                ifempsal*1.05>4000thenraisetoohighsal
                else update
                EMPsetSQL. ..endif ;
        exception
                whenNO DATAFOUND—
                notupleselectedthenrollback;
                whentoo_high_saltheninsertintohighsalempsvalues(empno);com
                mit;
        end;
```

Afterthekeywordwhenalistofexceptionnamesconnectedwithorcanbespecified.Thelast

whenclause in the exception part may contain the exception name others. This introduces the defaultexceptionhandlingroutine, forexample, arollback.

If a PL/SQL program is executed from the SQL*Plus shell, exception handling routines may containstatementsthatdisplayerrororwarningmessagesonthescreen.Forthis,theprocedureraiseapplication errorcanbeused.Thisprocedurehastwoparameters<errornumber>and<messagetext>. <errornumber>isanegativeintegerdefinedbytheuserandmustrange between-20000and-20999. <errormessage>isastringwithalengthupto2048characters.

Theconcatenationoperator“||”canbeusedtoconcatenatesinglestringstoonestring.Inorder

todisplaynumericvariables,thesevariablesmustbeconvertedtostringsusingthefunctiontochar.Ifthe procedure raise application error is called from a PL/SQL block, processing the PL/SQL blockterminatesandall databasemodificationsareundone,thatis,animplicit

rollbackisperformedinadditionto displayingtheerror message.

Example:

```
        ifempsal*1.05>4000
        thenraiseapplicationerror(-20010,�because Salaryincreaseforemployeewithld‟||tochar(EMPno)||‟istoo
        high‟);
```

E.g.

```
        Declare
                V_maxnonumber(2):=20;
                V_curno number
                (2);E_too_many_empexcepti
                on;
        Begin
                Selectcount(empno)intov_curnofromempW
                heredeptno=10;
                Ifv_curno>25thenRaise
                e_too_many_Emp;End
                if;
        Exception
                whene_too_many_empthen
                ….
                …..
        end;
```

LabExercise

1) Writea PL/SQLblocktocalculate factorial. UseException Handling.
2) Writea PL/SQLblocktofindprime numberforfirst30 numbers.
3) Writea PL/SQLblocktofindFibonacciseries for first50 numbers.
4) Writea PL/SQLblocktofind**a**raised topower **bi.e. a**^{b}
5) Writea PL/SQLblocktofindthe grade of astudent. Entermarks for5subjects.
6) Writea PL/SQLblocktoupdate thetable.**Table:ACCT_MSTR==>**

ACCT_NO	CURBAL
SB1	500
SB5	500
SB9	500
SB13	500

7) Writeonyourownone PL/SQLblockfortheproblemstatement.

FAQ:

1) WhatisPL/SQL? Explain.
2) Whatisthedifferencebetween"SQL"and"PL/SQL"?
3) WhatarethedifferentGoalsofPL/SQL?
4) Whatareexceptions?Whatarethedifferenttypesofexceptions?
5) WhatarethedifferentconditionalstatementsusedinPL/SQL?
6) WhatarethedifferentiterativeconstructusedinPL/SQL?Explaininshort.
7) WhatarethefeaturesofPL/SQL?Explain.
8) WhataretheadvantagesofPL/SQL?Explain
9) Howwillyoustopaninfiniteloopwithoutclosingtheprogram?
10) WhyPL/SQLdoesnotsupportretrievingmultiple records?

AssignmentNo.	**5**
Title	Write aPL/SQLstoredprocedure andfunction.
PROBLEM STATEMENT/DEFINITION	WriteaStoredProcedurenamelyproc_Gradeforthecategorizationofstudent.Ifmarksscoredbystudentsinexaminationis<=1500andmarks>=990thenstudentwillbeplacedindistinctioncategory if marks scored are between 989 and900 category is first class, if marks899and 825categoryisHigherSecondClass.
Objectives	• Understand IF-THEN condition and FOR loop • Understand the PL/SQL Stored Procedure. • Understand the PL/SQL Stored Function • Write PL/SQL block code using stored procedure and stored function.
Software packages and hardware apparatus used	MySQL/Oracle PC with the configuration as Latest Version of 64 bit Operating Systems, Open Source Fedora-GHz. 8 G.B. RAM, 500 G.B. HDD, 15"Color Monitor, Keyboard, Mouse
References	Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", 5th Edition, McGraw Hill Publishers, 2002, ISBN 0-07-120413-X Connally T., Begg C., "Database Systems", 3rd Edition, Pearson Education, 2002, ISBN 81-7808-861-4 mysql.com/docs/mysql-tutorial-excerpt-5.1-en.pdf
STEPS	Refer to details
Instructions for writing journal	• Date • Title • Problem Definition • Learning Objective • Learning Outcome • Theory-Related concept,Architecture,Syntax etc • Class Diagram/ER diagram • Test cases • Program Listing • Output • Conclusion

AssignmentNo:5

Title :- PL/SQL Stored Procedure and Stored Function Write a Stored Procedure namely proc_Gradeforthecategorizationofstudent.Ifmarksscoredbystudentsinexaminationis<=1500andmarks>= 990then student will be placed in distinction category if marks scored are between 989 and900 category isfirst class, if marks 899 and 825 category is Higher Second Class. Write a PL/SQL block for usingprocedure created with above requirement. Stud_Marks(name, total_marks) Result(Roll,Name, Class).**Frame the separate problem statement for writing PL/SQL Stored Procedure and function, inlinewithabovestatement.Theproblemstatementshouldclearlystatetherequirements.**

Objective:-a) Understand IF-THEN condition and FOR loop b)Understand the PL/SQL Stored Procedure c)Understand the PL/SQL Stored Function d)Write PL/SQL block code.

Theory :-

PROCEDURE:-

Aprocedure isa subprogramthatperforms aspecific action ortask.Aprocedurehastwo parts.

1) Theprocedurespecification:Theprocedurespecificationspecifiestheprocedurenameandtheparametersitaccepts.Itisnot necessarytocreateaprocedurethatacceptsparameters.

2) Theprocedurebody:Theprocedurebodycontainsthedeclarativesectionwithout DECLAREkeyword,theexecutablesection andanexception section.

Syntaxforcreatingaprocedure

Create[orreplace]PROCEDUREprocedure_name[(

argument1 [IN / OUT / IN OUT] type),

(argument2[IN/OUT/INOUT]type),

….]

IS/AS

Procedure_body

Where

Procedure_name: –

isthenameoftheproceduretobecreatedArgument:-

isthenameoftheprocedureparameter

Type:-Isthedatatypeoftheassociatedparameter

Procedure_body:-Isa PL/SQLblockthatmakes up thecode oftheprocedure.

IN:-

Thisisdefaultmode.Thevalueoftheactualparameterispassedintotheprocedure.Insidetheprocedurethe formal parameterisconsideredread only.

OUT:-

Anyvaluetheactualparameterhaswhentheprocedureiscalledignored.Insidetheprocedure,thefor mal

para

met

ersa

reco

nsid

ered

aswr

iteo

nly.

INOUT:-
thismodeiscom
binationofINa
ndOUT

Deletingprocedure:-Toremoveaprocedurefromthedatabase.

Syntax:-

Dropprocedure<procedure_name>;

FUNCTION:-

A function is asubprogram, which is usedto compute values. It is similar to a procedure, function alsotakes arguments and can be in different modes. Function also can be stored in the database. It is aPL/SQLblockconsistingofdeclarative, executableandexceptionsection.

Difference betweenprocedure andfunction isthattheprocedurecallis aPL/SQLstatementbyitself,whileafunction calliscalled as apart of an expression.

Afunctioncanreturnmore thanone value usingOUTparameter.Afunctioncanbecalled usingpositionalornamed notation.

Syntaxforcreatingafunction:-

```
Create[orreplace]FUNCTIONfunction_name[(
argument1 [IN / OUT / IN OUT] type),
(argument2[IN/ OUT/INOUT]type),
….]
Returnreturn_type IS/AS

Function_body
```

Where

Function_name:–

isthenameofthefunctiontobecreatedArgument:-

isthenameofthefunctionparameter

Type:-Isthedatatypeoftheassociatedparameter

Function_body:-IsaPL/SQLblockcontaining code forthe function.

IN:-

Thisisdefaultmode.Thevalueoftheactualparameterispassedintotheprocedure.Insidetheprocedurethe formal parameterisconsideredread only.

OUT:-

Anyvaluetheactualparameterhaswhentheprocedureiscalledignored.Insidetheprocedure,thefor mal parametersareconsideredaswriteonly.

INOUT:-thismodeiscombinationofINandOUT

DeletingaFunction:-Toremovethesubprogramfromthedatabase.

Syntax:-

Dropfunction<function_name>;

Package:

A package is a PL/SQL construct that allows related objects to be stored together. A package has 2separateparts:thespecificationandthebody.Eachofthemstoredseparatelyinthedatadictionary.**PackageSpecification**:

```
CREATEORREPLACEPACKAGEpackage_name
{IS|AS}
type_definition|
procedure_specification|Functionspecification|
variable_declaration|
exception_declaration |
cursor_declaration |
pragma declaration |
end[procedure_name];
```

PackageBody:

Thepackagebodyisseparatedatadictionaryobjectfromthepackageheader.Itcannotbesuccessfullycompiled unlessthepackageheaderhasalreadybeensuccessfullycompiled.

Syntax:

```
CREATEORREPLACEPACKAGEBODYpackage_nameASProced
uredefinition;
Functiondefinition;
…….
Endpackage_name
```

To drop the package(both specification & the body) use the **drop package** command as

follows:Syntax :

Droppackage<package_name>;

LabExercise

1) WriteaprocedureonEMPtable.Itshouldincreasecommissionofanemployee.Employeenumberandcommissionarepassedasparameterstothecalled procedure.
2) Writeafunctionthatreturnsthenumberofemployeesworkinginadepartment.Passdepartmentnumber as an inputto thefunction.
3) Createtableclasseswiththefollowingfields (Deptno,course,cur_student,max_student)Insert4or5recordsand

Write a function which returns true if the specified class is 80 percent full or more, and falseotherwise.Write a PL/SQLblock tocallthis functionand use cursorin PL/SQLblock to holdtherecords of alldepartment.

4) Write a procedure to update records of classes table and write a PL/SQL block to call thatprocedure.
5) Createapackagewhichconsistofproceduresforinsert,deleteandupdatethedataofclassestable.

FAQ:

1) Explainthe termprocedure and functionof PL/SQLin short.
2) Whatisthedifferencebetween"procedure"and"function"?
3) Whatisthedifferencebetween"%type"and"%rowtype"?
4) Whatispackage?Explain.
5) Whatistheuseofpackage?
6) Whatarethedifferentmodesofargumentpassing?
7) WhatisdifferencebetweenIN&INOUT?
8) Writeapackagewhichconsistsofcursor,trigger,procedure&function.
9) Whataretheadvantagesofprocedure&function?
10) Writethesyntaxtodropfunction,procedure&package.

AssignmentNo.	**6**
Title	Cursors: (Alltypes:Implicit,Explicit,CursorFORLoop,ParameterizedCursor)
PROBLEM STATEMENT/DEFINITION	Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table
Objectives	• Understand the types of cursors • Understand how to use cursors with PL/SQL block
Software packages and hardware apparatus used	MySQL/Oracle PC with the configuration as Latest Version of 64 bit Operating Systems, Open Source Fedora-GHz. 8 G.B. RAM, 500 G.B. HDD, 15"Color Monitor, Keyboard, Mouse
References	Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", 5th Edition, McGraw Hill Publishers, 2002, ISBN 0-07-120413-X Connally T., Begg C., "Database Systems", 3rd Edition, Pearson Education, 2002, ISBN 81-7808-861-4 mysql.com/docs/mysql-tutorial-excerpt-5.1-en.pdf
STEPS	Refer to details
Instructions for writing journal	• Date • Title • Problem Definition • Learning Objective • Learning Outcome • Theory-Related concept,Architecture,Syntax etc • Class Diagram/ER diagram • Test cases • Program Listing • Output • Conclusion

AssignmentNo:6

Title :-Cursors: (All types: Implicit, Explicit, Cursor FOR Loop, Parameterized Cursor) Write aPL/SQL block of code using parameterized Cursor, that will merge the data available in the newlycreated table Cust_New with the data available in the table Cust_Old. If the data in the first tablealreadyexistinthesecondtablethenthatdatashouldbeskipped.Frametheseparateproblemstatementforw ritingPL/SQLblocktoimplementalltypes

Objective :-a)Understand the types of cursors b) Understand how to use cursors with PL/SQL block

Theory :-

CURSOR:-For theprocessingof anySQLstatement, databaseneeds toallocatememory.Thismemoryiscalledcontext area. The context area is a part of PGA (Process global area) and is allocated on the oracle

server.Acursorisassociatedwiththisworkarea used byORACLE, for multirowqueries.Acursor is ahandleorpointertothecontextarea.Thecursorallowstoprocesscontentsinthecontextarearowbyrow. Therearetwotypesofcursors.

1) Implicitcursor:-

   ImplicitcursorsaredefinedbyORACLEimplicitly.ORACLEdefinesimplicitcursorfor

   everyDMLstatements.

2) Explicitcursor:-Theseareuser-definedcursorswhicharedefinedinthedeclarationsectionofthePL/

   SQLblock.Thereare foursteps inwhich theexplicitcursor is processed.

   1) Declaringacursor
   2) Openingacursor
   3) Fetchingrowsfromanopenedcursor
   4) Closingcursor

GeneralsyntaxforCURSOR:-

```
DECLARE
        Cursorcursor_nameISselect_statementorquery;B
EGIN
        Opencursor_name;
Fetchcursor_nameintolist_of_variables;C
        losecursor_name;
END;
```

Where

1) Cursor_name:-isthenameofthecursor.
2) Select_statement:-isthequerythatdefinesthesetofrowstobeprocessedbythecursor.
3) Opencursor_name:-

   openthecursorthathasbeenpreviouslydeclared.Whencursor is openedfollowingthings happen

i) Theactivesetpointerissettothefirstrow.

ii) Thevalueofthebindingvariablesareexamined.

4) Fetchstatementisusedtoretrievearowfromtheselectedrows,oneatatime,intoPL/ SQLvariables.

5) Closecursor_name:-Whenallofcursorrowshavebeenretrieved,thecursorshouldbeclosed.

Explicitcursorattributes:-

Followingarethecursorattributes

1. %FOUND:-

ThisisBooleanattribute.ItreturnsTRUEifthepreviousfetchreturnsarowandfalseifitdoesn‟t.

2. %NOTFOUND:-If fetch returns a row it returns FALSE and TRUE if it isoftenused doesn‟t. Thisastheexitcondition for thefetch loop;

3. %ISOPEN:-Thisattributeisusedtodeterminewhetherornot theassociatedcursorisopen.IfsoitreturnsTRUEotherwiseFALSE.

4. %ROWCOUNT:-Thisnumericattributereturnsanumberofrowsfetchedbythecursor.

CursorFetchLoops

1) Simple

LoopSyntax:-

```
LOOP
    Fetchcursornameintolistofvariables;
EXIT WHEN cursorname
    %NOTFOUNDSequence_of_stat
    ements;
ENDLOOP;
```

2) WHILE

LoopSyntax:-

```
FETCHcursornameINTOlistofvariables;W
HILEcursorname%FOUNDLOOP
    Sequence_of_statements;
    FETCHcursornameINTOlistofvariables;E
NDLOOP;
```

3) Cursor

FORLoopSyn

tax:

```
FORvariable_nameINcursornameLOOP
    --animplicitfetchisdonehere.
    --cursorname%NOTFOUNDisalsoimplicitlychecked.
```

```
                    --
                    processthefetchrecords.Seq
                    uence_of_statements;
                ENDLOOP;
```

Therearetwoimportantthingstonoteabout:-

i) Variable_nameisnotdeclaredintheDECLAREsection.Thisvariableisimplicitlydeclar edbythePL/SQLcompiler.

ii) Typeofthisvariableiscursorname%ROWTYPE.

ImplicitCursors

PL/SQLissuesanimplicit cursorwheneveryouexecute a SQLstatementdirectlyin your code,as longas that code does not employ an explicit cursor. It is called an "implicit" cursor because you, thedeveloper,donot explicitlydeclare acursor for theSQLstatement.

If you use an implicit cursor, Oracle performs the open, fetches, and close for you automatically; theseactionsareoutsideofyourprogrammaticcontrol.Youcan,however,obtaininformationaboutthemostre centlyexecutedSQLstatementbyexaminingthevalues intheimplicitSQLcursor attributes.

PL/SQL employs an implicit cursor for each UPDATE, DELETE, or INSERT statement you execute in aprogram.Youcannot,inotherwords, executethesestatementswithinanexplicit cursor,evenifyouwantto.Youhaveachoicebetweenusinganimplicitorexplicitcursoronlywhenyouexecuteasingle-rowSELECTstatement(aSELECTthatreturnsonlyonerow).

In the following UPDATE statement, which gives everyone in the company a 10% raise, PL/SQLcreatesanimplicitcursortoidentifythesetofrowsinthetablewhichwouldbeaffectedbytheupdate:

```
UPDATEemployee
   SETsalary=salary*1.1;
```

Thefollowingsingle-rowquerycalculatesandreturnsthetotalsalaryforadepartment.Onceagain,PL/ SQLcreatesan implicitcursorforthis statement:

```
SELECTSUM(salary)INTOdepartment_totalF
  ROMemployee
 WHEREdepartment_number=10;
```

If you have a SELECT statement that returns more than one row, you must use an explicit cursor forthatqueryandthenprocessthe rowsreturnedoneatatime. PL/SQLdoes notyetsupportanykind ofarrayinterface betweena database tableanda composite PL/SQLdatatypesuch as aPL/SQLtable.

DrawbacksofImplicitCursors

Evenifyourqueryreturnsonlyasinglerow,youmightstilldecidetouseanexplicitcursor.Theimplicit cursorhas thefollowingdrawbacks:

- Itislessefficientthananexplicitcursor
- Itismorevulnerabletodataerrors
- Itgivesyoulessprogrammaticcontrol

Thefollowingsectionsexploreeachoftheselimitationstotheimplicitcursor.

Inefficienciesofimplicitcursors

An explicit cursor is, at least theoretically, more efficient than an implicit cursor. An implicit cursorexecutes as a SQL statement and Oracle's SQL is ANSI-standard. ANSI dictates that a single-row querymust not only fetch the first record, but must also perform a second fetch to determine if too many rowswill be returnedbythatquery(sucha situationwillRAISEtheTOO_MANY_ROWS PL/SQLexception).Thus,animplicitqueryalwaysperformsaminimumoftwofetches,whileanexplicitcursoronlyneedstoperformasinglefetch.

Thisadditionalfetchisusuallynotnoticeable,andyoushouldn'tbeneuroticaboutusinganimplicitcursor for a single-row query (it takes less coding, so the temptation is always there). Look out forindiscriminate use of the implicit cursor in the parts of your application where that cursor will beexecutedrepeatedly.Agoodexample isthe Post-Querytriggerin theOracleForms.

Post-Query fires once for each record retrieved by the query (created from the base table block and thecriteriaenteredbytheuser).Ifaqueryretrievestenrows,thenanadditionaltenfetchesareneededwithan implicit query. If you have 25 users on your system all performing a similar query, your server mustprocess 250 additional (unnecessary) fetches against the database. So, while it might be easier to writean implicit query, there are some places in your code where you will want to make that extra effort andgowith theexplicitcursor.

Vulnerabilitytodataerrors

If an implicit SELECT statement returns more than one row, it raises the TOO_MANY_ROWSexception. When this happens, execution in the current block terminates and control is passed to theexceptionsection.Unlessyoudeliberatelyplantohandlethisscenario,useoftheimplicitcursorisadeclarationoffaith.Youaresaying,"Itrustthat querytoalwaysreturnasinglerow!"

Itmaywellbethattoday,withthecurrentdata,thequerywillonlyreturnasinglerow.Ifthenatureofthedataeverchanges,however,youmayfindthattheSELECTstatementwhichformerlyidentifiedasingle row now returns several. Your program will raise an exception. Perhaps this is what you willwant. On the other hand, perhaps the presence of additional records is inconsequential and should beignored.

With the implicit query, you cannot easily handle these different possibilities. With an explicit query,yourprogramwillbeprotectedagainstchangesindataandwillcontinuetofetchrowswithoutraisingexceptions.

LabExercise

1) Createtablewithname**student**havingthefield**rollno**,**firstname**,**lastname**&**branch**.Insert10 records into table. Write a PL/SQL to create a cursor to hold all the record of student tablehavingbranch„**ComputerScience**‟. Displayalltherecords.

2) Write a PL/SQLblocktoupdate therecord of rollno=100 &setthe**branch** to **EandTC'**,ifitis not present then insert the record into the student table with the id=100; (use implicit cursorsql %notfound).
3) Writeacursoranduseittoraisetheemployeesalariesasfollows:
   i) Allemployeesofdepartment20get5%raise
   ii) Allemployeesofdepartment30get10%raise
   iii) Restofemployeesget7.5%raiseUs
   eseparatecursor.

FAQ:

1) Whatiscursor?
2) Whatarethedifferenttypesofcursors?
3) Whatarethedifferentattributesofexplicitcursor?Explaininbrief.
4) Whatis implicitcursor?
5) ExplaintheFORloopofCursor.
6) Whatisdifferencebetweensimpleloop,whileloop&forloop?
7) WhatisdifferencebetweenImplicit&ExplicitCursor?
8) ExplainFORUPDATEcursorwithanexample.
9) WhatisCURRENTOFclauseincursor?Giveanexample.
10) Listallpredefinedcursor.

AssignmentNo.	7
Title	Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers).
PROBLEM STATEMENT/DEFINITION	Write a database trigger on Library table. The System should keep track of the records that arebeing updated or deleted. The old value of updated or deleted records should be added inLibrary_Audittable
Objectives	• Understand the concept of row level and statement level trigger • Understand the concept of trigger initiated against event.
Software packages and hardware apparatus used	MySQL/Oracle PC with the configuration as Latest Version of 64 bit Operating Systems, Open Source Fedora-GHz. 8 G.B. RAM, 500 G.B. HDD, 15"Color Monitor, Keyboard, Mouse
References	Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", 5th Edition, McGraw Hill Publishers, 2002, ISBN 0-07-120413-X Connally T., Begg C., "Database Systems", 3rd Edition, Pearson Education, 2002, ISBN 81-7808-861-4 mysql.com/docs/mysql-tutorial-excerpt-5.1-en.pdf
STEPS	Refer to details
Instructions for writing journal	• Date • Title • Problem Definition • Learning Objective • Learning Outcome • Theory-Related concept,Architecture,Syntax etc • Class Diagram/ER diagram • Test cases • Program Listing • Output • Conclusion

AssignmentNo:7

Title :- Database Trigger (All Types: Row level and Statement level triggers, Before and AfterTriggers).WriteadatabasetriggeronLibrarytable.TheSystemshouldkeeptrackoftherecordsthatare being updated or deleted. The old value of updated or deleted records should be added inLibrary_Audittable.

Objective **:-**a)Understand the concept of row level and statement level trigger

b)Understand the concept of trigger initiated against event

Theory **:-**

DATABASETRIGGERS:-

A database trigger is a PL/SQL program unit, which gets fired automatically whenever the data eventsuch as DML or DDL system event. Triggers are associated with a specific table and are firedautomaticallywhenever thetablegetsmanipulatedinapredefinedway.Theactofexecutingatriggeriscall edasfiringatrigger.

Triggers are similar to procedures in that they are named PL/SQL blocks with declarative, executableand exception handling sections. But the difference is a procedure is executed explicitly from anotherblockvia aprocedurecallbutatriggerisexecutedimplicitlywheneverthetriggeringeventhappens.A procedure canpassargumentsbuttriggerdoesn‟t acceptarguments

Adatabasetriggerhasfollowing components:-

1.Atriggering **Event**

2.Atriggering **Constraint**

3.A triggering

ActionTriggercategories

Triggersarecategorizedinvariousways.

1)Trigger

type2)Triggering

time3)Triggeringe

vent

Triggertypes

Therearetwotypesoftriggers

1. **Statement Trigger**:-A statement trigger is a trigger in which the trigger action is executed once forthemanipulation operationthatfires thetrigger.

2. **Row Trigger**:-A row trigger is a trigger in which the trigger action is performed repeatedly for eachrowofthetablethat isaffectedbythemanipulationoperationthatfiresthetrigger.

Triggeringtime

Triggerscanspecifythetimeoftriggeraction.

1) Beforethetriggeringevent

Thetriggeractionisperformedbeforetheoperationthatfiresthetriggerisexecuted.Thistriggerisusedwhen executionofoperationdepends ontrigger action.

2)Afterthetriggeringevent

Thetriggeractionisperformedaftertheoperationthatfiresthetriggerisexecuted.This trigger is used when triggering action depends on the execution of operation.**TriggeringEvents**

Triggering events are the DML operations. These operations are **insert, update and delete** When theseoperationsareperformed onatable,thetriggerwhichisassociatedwiththeoperationisfired. Triggeringeventsdividetriggersintothreetypes.

1) DELETETRIGGER
2) UPDATETRIGGER
3) INSERTTRIGGER

GeneralsyntaxforcreationofTrigger

```
Create[orreplace]TRIGGER<trigger_name>
<BEFORE|AFTER>
DELETE| [OR]INSERT|[OR] UPDATE[OF<column1>[,<column2>…..]
ON<table_name>
[foreachrow[when<condition>]B
egin
………        ………
……………….
End;
```

Where

Trigger_name:-trigger name is the name of the trigger.Table_name:-
isthyetablenameforwhichthetriggerisdefined.
Trigger-condition:-
Thetriggerconditioninthewhenclause,ifpresentisevaluatedfirst.Thebodyofthetriggeris executed onlywhen thiscondition evaluates totrue.

Droppingtrigger

Supposeyouwanttodroptriggerthenthesyntax is

Syntax:-Droptriggertrigger_name;

EnablingandDisablingTriggers

TheTriggercanbedisabledwithoutdroppingthem.Whenthetriggerisdisabled,itisstillexistsindatadictionarybut neverfired,Todisabletrigger, usealtercommand.

Syntax:-

AlterTRIGGERtrigger_nameDISABLE/

ENABLE;Foralltriggerson aparticulartable

Syntax:-

AlterTRIGGERtrigger_name(DISABLE/ENABLE)alltriggers;

LabExercise:-

1) CreateatriggerthatauditstheoperationsonanEmptable.Steps

   Createtableemp_audit

   (idnumber,operationvarchar2(6),Dtdate,User_idnumber,Usernamevarchar2(20));

   Ifanyoperationlikeinsert,update,deletedoneonEMPtabletheninsertintoEMP_audittableinformationlikethenameof theoperation with id, user_id anddate.

2) CreateatableEmployee(id,Emp_name,Salary,City)

   CreateatriggertoconverttheEmp_nameintouppercasebeforeinsertingorupdatingonEmployeetable.

3) CreateatriggertocheckSalaryislessthan20000beforeinsertingorupdatingonEmployeetable.

4) Createatrigger(StatementLevelTrigger)todisplaymessagesafterinsertingorupdatingordeletingrecords onEmployeeTable.

FAQ:

1) WriteadatabaseTrigger
2) ExplainDatabaseTriggerComponents.
3) ExplainTriggerTypeswithe.g.
4) ExplaindifferencebetweenRow-Level&Statement-LevelTrigger.
5) WriteaSyntaxforEnable&DisableTrigger.
6) WriteaSyntaxforDisplayingTriggerErrors.

AssignmentNo.	**8**
Title	**DatabaseConnectivity:**
PROBLEM STATEMENT/DEFINITION	WriteaprogramtoimplementMySQL/ Oracledatabaseconnectivitywithanyfrontendlanguagetoimplement Databasenavigationoperations(add,delete,editetc.)
Objectives	Insert a record in mysql database using Java/PHP. update a record in mysql database using Java/PHP.
Software packages and hardware apparatus used	MySQL/Oracle PC with the configuration as Latest Version of 64 bit Operating Systems, Open Source Fedora-GHz. 8 G.B. RAM, 500 G.B. HDD, 15"Color Monitor, Keyboard, Mouse
References	Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", 5th Edition, McGraw Hill Publishers, 2002, ISBN 0-07-120413-X Connally T., Begg C., "Database Systems", 3rd Edition, Pearson Education, 2002, ISBN 81-7808-861-4 http://docs.mongodb.org/manual
STEPS	Refer to details
Instructions for writing journal	• Date • Title • Problem Definition • Learning Objective • Learning Outcome • Theory-Related concept,Architecture,Syntax etc • Class Diagram/ER diagram • Test cases • Program Listing • Output • Conclusion

AssignmentNo.	**9**
Title	**MongoDBQueries:** DesignandDevelopMongoDBQueriesusingCRUDoperations.(UseCRUDoperations, SAVEmethod,logicaloperators)
PROBLEM STATEMENT/DEFINITION	DesignandDevelopMongoDBQueriesusingCRUDoperations.(UseCRUDoperations,SAVEmethod,logicaloperatorsetc.).
Objectives	• Understand the concept of Binary JSON format. • Understand the concept of Mongo DB document model.
Software packages and hardware apparatus used	MongoDB PC with the configuration as Latest Version of 64 bit Operating Systems, Open Source Fedora-GHz. 8 G.B. RAM, 500 G.B. HDD, 15"Color Monitor, Keyboard, Mouse
References	1.Kristina Chodorow,Michael Dierolf,"MongoDB:The definite Guide", O'Reilly Publications,ISBN:978-1-449-34468-9. 2.Kevin Roebuck,"Storing and Managing Big Data-NoSQL,Hadoop and More",Emereopty Limited,ISBN:1743045743,9781743045749 http://docs.mongodb.org/manual
STEPS	Refer to details
Instructions for writing journal	• Date • Title • Problem Definition • Learning Objective • Learning Outcome • Theory-Related concept,Architecture,Syntax etc • Class Diagram/ER diagram • Test cases • Program Listing • Output • Conclusion

AssignmentNo.09

Aim **:** DesignandDevelopMongoDBQueriesusingCRUDoperations. (UseCRUDoperations,SAVEmethod,logicaloperators)

- **Objectives** **:**Understand the concept of Binary JSON format.
  Understand the concept of Mongo DB document model.

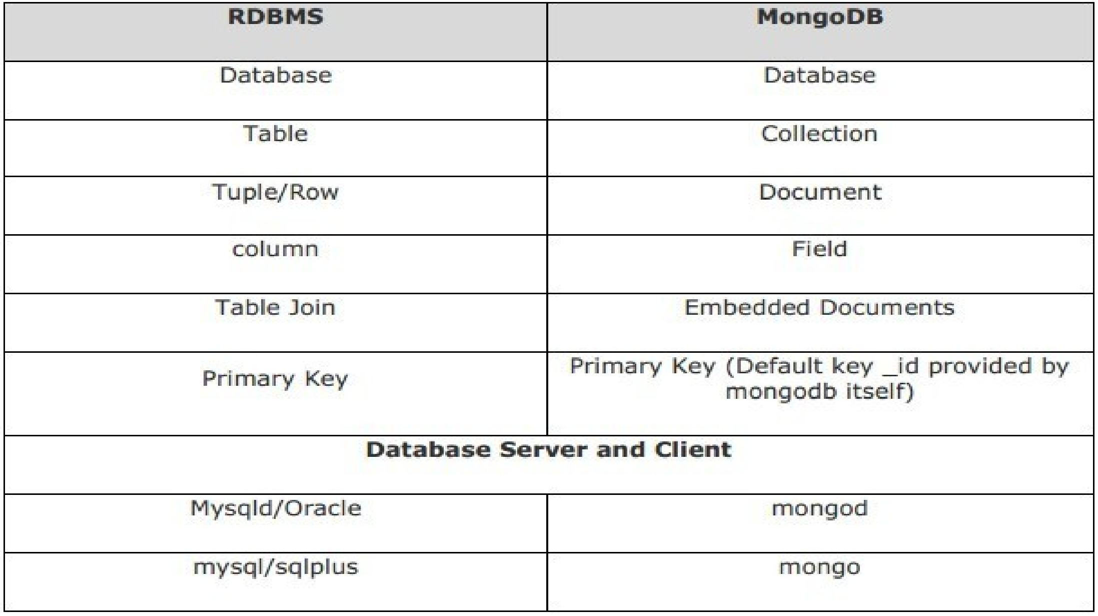
Theory :**MongoDB** is a cross-platform, document oriented database that provides, highperformance,highavailability,andeasyscalability.MongoDBworksonconceptofcollectionanddocument.

Database
Databaseisaphysicalcontainerforcollections.Eachdatabasegetsitsownsetoffilesonthefilesystem.Asingle MongoDBservertypicallyhas multipledatabases.

Collection
Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collectionexistswithinasingledatabase.Collectionsdonotenforceaschema.Documentswithinacollectioncanhavedifferentfields.Typically,alldocumentsinacollectionareofsimilarorrelatedpurpose.

Document
A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means thatdocuments in the same collection do not need to have the same set offields or structure, and commonfieldsin acollection's documentsmayholddifferenttypesofdata.

ThefollowingtableshowstherelationshipofRDBMSterminologywithMongoDB.



RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by mongodb itself)
Database Server and Client	
Mysqld/Oracle	mongod
mysql/sqlplus	mongo

CRUDisthebasicoperationofMongodb,itstandsCREATE,READ,UPDATE,DELETE.

MongoDB─1. Create Collection

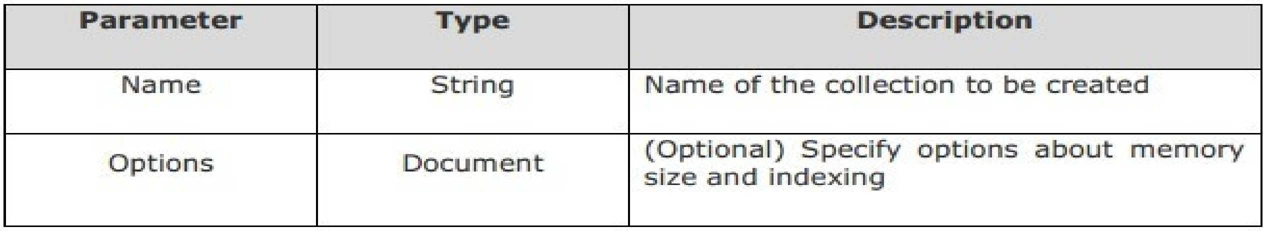
ThecreateCollection()Method

MongoDBdb.createCollection(name, options) isusedto

createcollection.Basic syntax of createCollection() command is as

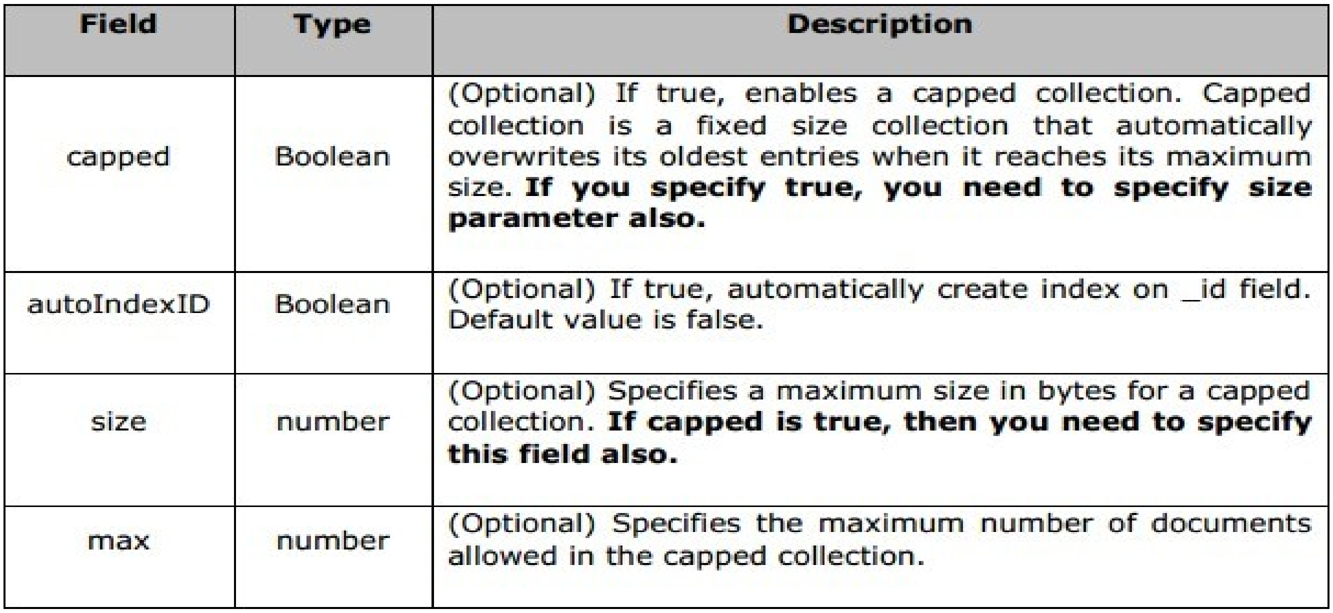
follows:**db.createCollection(name,options)**

In the command, name is name of collection to be created. Options are a document and are used to specifyconfigurationof collection.



Parameter	Type	Description
Name	String	Name of the collection to be created
Options	Document	(Optional) Specify options about memory size and indexing

Options parameter is optional, so you need to specify onlythename of the collection. Following is the list ofoptions youcanuse:



Field	Type	Description
capped	Boolean	(Optional) If true, enables a capped collection. Capped collection is a fixed size collection that automatically overwrites its oldest entries when it reaches its maximum size. **If you specify true, you need to specify size parameter also.**
autoIndexID	Boolean	(Optional) If true, automatically create index on _id field. Default value is false.
size	number	(Optional) Specifies a maximum size in bytes for a capped collection. **If capped is true, then you need to specify this field also.**
max	number	(Optional) Specifies the maximum number of documents allowed in the capped collection.

While inserting the document, MongoDB first checks size field of capped collection, then

itchecksmaxfield.

Examples

Basic syntaxofcreateCollection()method without options isas follows:

>use test

switched to db test

>db.createCollection("mycollection")

```
{ "ok" : 1}

>
```

Youcancheckthecreatedcollectionbyusingthecommandshowcollections.

```
>show

collectionsmycoll

ection

system.indexes
```

2. READ-Thefind()Method

ToquerydatafromMongoDBcollection,youneedtouseMongoDB'sfind()method.Syntax

Thebasicsyntaxoffind()methodisasfollows:

```
>db.COLLECTION_NAME.find()
```

find()method will display all the documents in a non-structured

way.Thepretty() Method

Todisplaytheresultsinaformattedway,youcanusepretty()method.Syntax

```
>db.mycol.find().pretty()
```

Example

```
>db.mycol.find().pretty()

{

"_id":

ObjectId(7df78ad8902c),"title":

"MongoDBOverview",

"description":"MongoDBisno

sqldatabase","by": "tutorials point",

"url":

"http://www.tutorialspoint.com","tags":

["mongodb", "database", "NoSQL"],"likes":

"100"

}

>
```

Apartfromfind()method,thereis findOne() method, thatreturnsonlyonedocument.

3. UPDATE

MongoDB'supdate()andsave()methodsareusedtoupdatedocumentintoacollection.

Theupdate()methodupdatesthevaluesintheexistingdocument whilethesave()methodreplacestheexistingdocumentwith thedocumentpassed insave()method.

MongoDBUpdate()Method

Theupdate()methodupdatesthevaluesintheexistingdocument.Theb

asicsyntaxofupdate() methodisas follows:

>db.COLLECTION_NAME.update(SELECTIOIN_CRITERIA,UPDATED_DATA)

Example

Considerthemycol collectionhasthefollowingdata.

{"_id":ObjectId(5983548781331adf45ec5),"title":"MongoDBOverview"}

{"_id":ObjectId(5983548781331adf45ec6),"title":"NoSQLOverview"}

{ "_id" : ObjectId(5983548781331adf45ec7), "title":"Tutorials Point

Overview"}Followingexamplewillsetthenewtitle'NewMongoDBTutorial'ofthedocumentsw

hosetitleis 'MongoDBOverview'.

>db.mycol.update({'title':'MongoDBOverview'},{$set:

{'title':'NewMongoDBTutorial'}})

>db.mycol.find()

{"_id":ObjectId(5983548781331adf45ec5),"title":"NewMongoDBTutorial"}

{"_id":ObjectId(5983548781331adf45ec6),"title":"NoSQLOverview"}

{"_id":ObjectId(5983548781331adf45ec7),"title":"TutorialsPointOverview"}

>

Bydefault,MongoDBwillupdateonlyasingledocument.Toupdatemultipledocuments,youne

ed toset aparameter'multi'totrue.

>db.mycol.update({'title':'MongoDBOverview'},

{$set:{'title':'NewMongoDBTutorial'}},{multi:true})

MongoDBSave()Method

Thesave()methodreplacestheexistingdocumentwiththenewdocumentpassedinthesave()method.

ThebasicsyntaxofMongoDBsave()methodis−

```
>db.COLLECTION_NAME.save({_id:ObjectId(),NEW_DATA})
```

Example

Followingexamplewillreplacethedocumentwiththe_id'5983548781331adf45ec7'.

```
>db.mycol.save(
{
"_id":ObjectId(5983548781331adf45ec7),"title":"TutorialsPointNewTo
pic",
"by":"TutorialsPoint"
} )
>db.mycol.find()
{"_id":ObjectId(5983548781331adf45ec5),"title":"TutorialsPointNewTopic","by
":"TutorialsPoint"}
{"_id":ObjectId(5983548781331adf45ec6),"title":"NoSQLOverview"}
{"_id":ObjectId(5983548781331adf45ec7),"title":"TutorialsPointOverview"}
```

4. DELETE-Theremove()Method

MongoDB'sremove()methodisusedtoremoveadocumentfromthecollection.

remove()methodacceptstwoparameters.OneisdeletioncriteriaandsecondisjustOneflag.

- deletioncriteria:(Optional)deletioncriteriaaccordingtodocumentswillberemoved.
- justOne:

(Optional)ifsettotrueor1,thenremoveonlyonedocument.Basicsyntaxofremove() methodis asfollows:

```
>db.COLLECTION_NAME.remove(DELLETION_CRITTERIA)
```

Example

Considerthemycol collectionhasthefollowingdata.

```
{"_id":ObjectId(5983548781331adf45ec5),"title":"MongoDBOverview"}
```

{"_id":ObjectId(5983548781331adf45ec6),"title":"NoSQLOverview"}

{ "_id" : ObjectId(5983548781331adf45ec7), "title":"Tutorials Point Overview"}Followingexamplewillremoveallthedocumentswhosetitleis'MongoDBOverview'.

>db.mycol.remove({'title':'MongoDBOverview'})

>db.mycol.find()

{"_id":ObjectId(5983548781331adf45ec6),"title":"NoSQLOverview"}

{"_id":ObjectId(5983548781331adf45ec7),"title":"TutorialsPointOverview"}

LOGICALOPERATORS:

ANDinMongoDB

Syntax

Inthefind()method,ifyoupassmultiplekeysbyseparatingthemby','thenMongoDBtreatsit asANDcondition.Following is thebasicsyntaxofAND –

>db.mycol.find({key1:value1,key2:value2}).pretty()

Example

Followingexamplewill showallthetutorialswrittenby'tutorialspoint'andwhosetitleis'MongoDBOverview'.

>db.mycol.find({"by":"tutorialspoint","title":"MongoDBOverview"}).pretty()

{

"_id":

ObjectId(7df78ad8902c),"title":

"MongoDBOverview",

"description":"MongoDBisnosqldatabase","b

y": "tutorialspoint",

"url":

"http://www.tutorialspoint.com","tags":

["mongodb","database","NoSQL"],"likes":"

100"

```
}>
```

For the above given example,equivalent where clause will be ' where by='tutorials point' AND title ='MongoDBOverview''.Youcanpassanynumberofkey,valuepairsinfindclause.

OR inMongoDB

Syntax : To query documents based on the OR condition, you need to use $or keyword.

Followingisthebasicsyntaxof OR−

```
>db.mycol.find({$or:[{key1:value1},{key2:value2}]}).pretty()
```

Examplewillshowallthetutorialswrittenby'tutorialspoint'orwhosetitleis'MongoDBOverview'.

```
>db.mycol.find({$or:[{"by":"tutorialspoint"},{"title":"MongoDBOverview"}]}).pretty()
```

```
{"_id":ObjectId(7df78ad8902c),"
title":"MongoDBOverview",
"description":"MongoDBisnosqldatabase","b
y": "tutorialspoint",
"url":
"http://www.tutorialspoint.com","tags":
["mongodb","database","NoSQL"],"likes":"
100"}
```

UsingANDandORTogetherExample

The followingexample will show the documents that have likes greater than 100 and whose title iseither 'MongoDB Overview' or by is 'tutorials point'. EquivalentSQL where clause is 'where likes>10AND(by='tutorialspoint'ORtitle='MongoDBOverview')'

```
>db.mycol.find({"likes": {$gt:10}, $or: [{"by": "tutorials point"},{"title":
                    "MongoDBOverview"}]}).pretty()
```

```
{
"_id":
ObjectId(7df78ad8902c),"title":
"MongoDBOverview",
"description":"MongoDBisnosqldatabase","b
y": "tutorialspoint",
"url":
"http://www.tutorialspoint.com","tags":
["mongodb","database","NoSQL"],"likes":"
100"}
```

Conclusion: ThuswehavestudiedMongoDBQueriesusingCRUDoperations.

FAQ:-

1. ExplainCREATEOperationwithexample.
2. ExplainANDOperatorwithexample.
3. ExplainDELETEfunctioninMongodb.
4. ExplainDELETEfunctioninMongodb.
5. ExplainFINDfunctioninMongodb.
6. ExplainOROperatorwithexample.

AssignmentNo.	**10**
Title	**MongoDBAggregationandIndexing:** Implementaggregationandindexingwithsuitableexample usingMongoDB.
PROBLEM STATEMENT/DEFINITION	DesignandDevelopMongoDBQueriesusingaggregationandindexing withsuitableexampleusingMongoDB
Objectives	Understand indexing and aggregation concept on MongoDB
Software packages and hardware apparatus used	Mongodb Operating Systems • (64-Bit)64-BIT Fedora 17 or latest 64-BIT Update of Equivalent Open source OS or latest 64-BIT Version and update of Microsoft Windows 7 Operating System onwards • Programming Tools (64-Bit) Latest Open source update of Eclipse Programming frame work, MongoDB 2.6.
References	**1.** MongoDB: The Definitive Guide, 2nd Edition,Powerful and Scalable Data Storage By Kristina Chodorow Publisher: O'Reilly Media **2.** http://docs.mongodb.org/manual
STEPS	Refer to details
Instructions for writing journal	• Date • Title • Problem Definition • Learning Objective • Learning Outcome • Theory-Related concept,Architecture,Syntax etc • Class Diagram/ER diagram • Test cases • Program Listing • Output • Conclusion

AssignmentNo.10

Aim : Implementaggregationandindexingwithsuitableexampleusing MongoDB.

Objectives : Understand indexing and aggregation concept on MongoDB records

Theory : MongoDB is an open-source document database and leading NoSQL database.MongoDB is written in C++. This tutorial will give you great understanding on MongoDB conceptsneeded tocreateanddeployahighlyscalableandperformance-orienteddatabase.

Aggregationsoperations process data records and return computed results. Aggregationoperations groupvalues from multiple documents together, and can perform a variety ofoperations on the grouped data to return asingleresult.InSQLcount(*) andwithgroupbyis anequivalentofmongodbaggregation.
The aggregate() Method For the aggregation in MongoDB, you should use aggregate()

method.Basicsyntaxof aggregate()methodis as follows:

>db.COLLECTION_NAME.**aggregate**(AGGREGATE_OPERATION)

Example

Inthe collectionyouhavethe followingdata:

{

_id:

ObjectId(7df78ad8902c)title:

'MongoDBOverview',

description: 'MongoDB is no sql

database',by_user:'tutorials point',

url:

'http://www.tutorialspoint.com',tags:

['mongodb','database','NoSQL'],likes:1

00

},

{

```
_id:ObjectId(7df78ad8902d)
```

```
title:'NoSQLOverview',
description: 'No sql database is very
fast',by_user:'tutorials point',
url:
'http://www.tutorialspoint.com',tags:
['mongodb','database','NoSQL'],likes:
10
},
{
_id:
ObjectId(7df78ad8902e)title:
'Neo4jOverview',
description: 'Neo4j is no sql
database',by_user:'Neo4j',
url:'http://www.neo4j.com',
tags:
['neo4j','database','NoSQL'],likes:
750
},
```

Now from the above collection, if you want to display a list stating how many tutorials are written by each user,thenyouwill use thefollowingaggregate()method:

```
>db.mycol.aggregate([{$group:{_id :"$by_user",num_tutorial:{$sum:1}}}])
{
"result":[
{
"_id": "tutorialspoint","num_tutorial":2
```

```
        },

        {
```

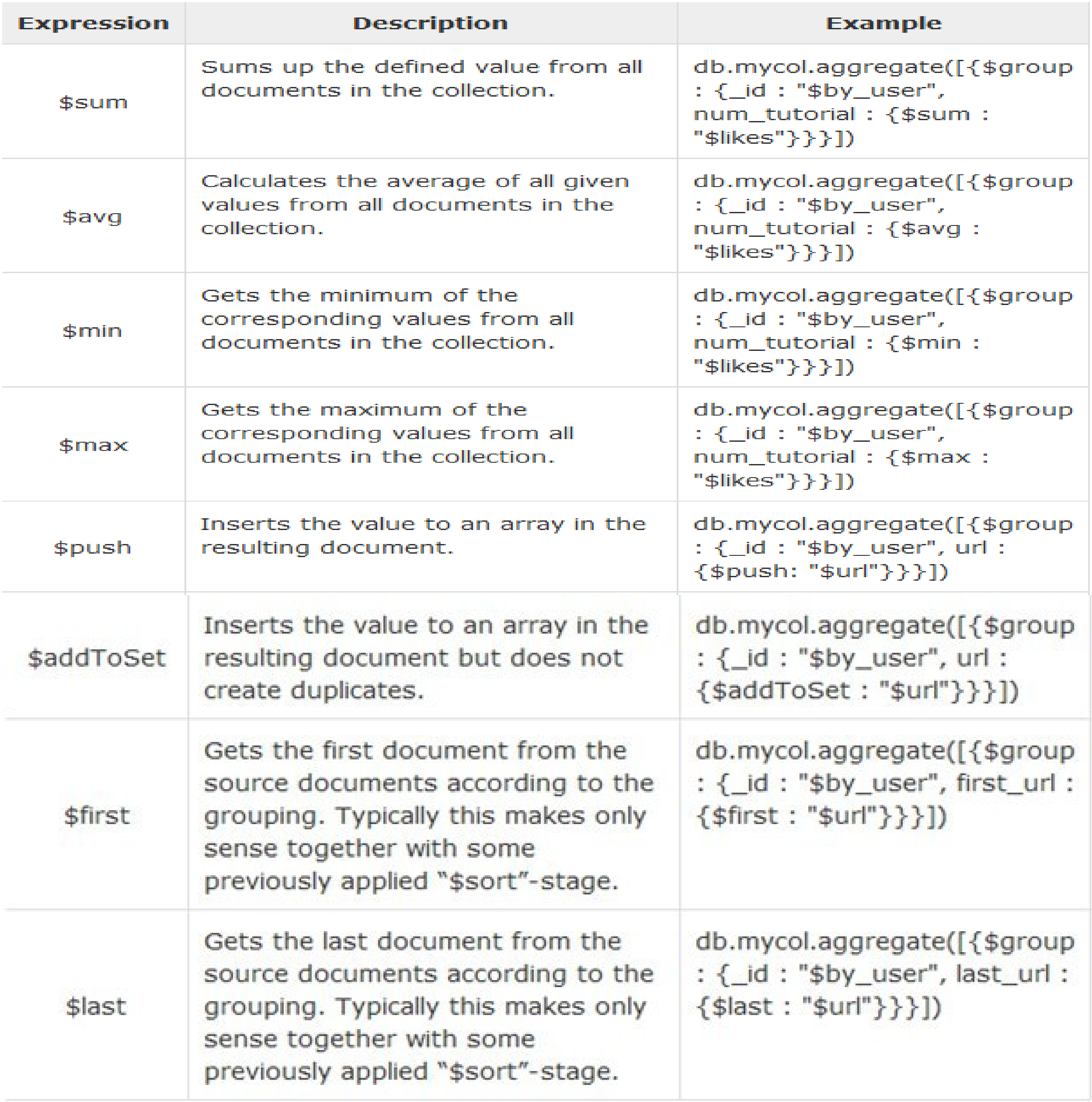
"_id": "Neo4j","num_tutorial":1

}],

"ok" : 1

}>

Sql equivalent query for the above use case will be select by_user, count(*) from mycol group byby_user.



Expression	Description	Example
$sum	Sums up the defined value from all documents in the collection.	db.mycol.aggregate([{$group : {_id : "$by_user", num_tutorial : {$sum : "$likes"}}}])
$avg	Calculates the average of all given values from all documents in the collection.	db.mycol.aggregate([{$group : {_id : "$by_user", num_tutorial : {$avg : "$likes"}}}])
$min	Gets the minimum of the corresponding values from all documents in the collection.	db.mycol.aggregate([{$group : {_id : "$by_user", num_tutorial : {$min : "$likes"}}}])
$max	Gets the maximum of the corresponding values from all documents in the collection.	db.mycol.aggregate([{$group : {_id : "$by_user", num_tutorial : {$max : "$likes"}}}])
$push	Inserts the value to an array in the resulting document.	db.mycol.aggregate([{$group : {_id : "$by_user", url : {$push: "$url"}}}])
$addToSet	Inserts the value to an array in the resulting document but does not create duplicates.	db.mycol.aggregate([{$group : {_id : "$by_user", url : {$addToSet : "$url"}}}])
$first	Gets the first document from the source documents according to the grouping. Typically this makes only sense together with some previously applied "$sort"-stage.	db.mycol.aggregate([{$group : {_id : "$by_user", first_url : {$first : "$url"}}}])
$last	Gets the last document from the source documents according to the grouping. Typically this makes only sense together with some previously applied "$sort"-stage.	db.mycol.aggregate([{$group : {_id : "$by_user", last_url : {$last : "$url"}}}])

PipelineConcept

InUNIXcommand,shellpipelinemeansthepossibilitytoexecuteanoperationonsomeinputandusetheoutput astheinput forthenext commandand soon.MongoDBalsosupportssameconcept in aggregation framework. There is a set ofpossible stages andeach of those is taken as a set ofdocumentsasaninputandproducesaresultingsetofdocuments(orthefinalresultingJSONdocumentat theendofthepipeline).This can thenin turnbeused forthenextstageand soon.

Followingarethepossiblestagesinaggregationframework:

- $project:Usedtoselectsomespecificfieldsfromacollection.
- $match:Thisisafilteringoperationandthusthiscanreducetheamountofdocumentsthataregivenasinputtothenextstage.

$group:Thisdoestheactualaggregationasdiscussedabove.

- $sort:Sortsthedocuments.
- $skip:Withthis,itispossibletoskipforwardinthelistofdocumentsforagivenamountofdocuments.
- $limit:Thislimitstheamountofdocumentstolookat,bythegivennumberstartingfromthecurrent positions.
- $unwind: This is used to unwind document that are using arrays. When using an array, the data iskindofpre-joinedandthisoperationwillbeundonewiththistohaveindividualdocumentsagain.Thuswiththis stagewewillincreasetheamountofdocuments forthenextstage.

Indexes support the efficient resolution of queries. Without indexes, MongoDB must scan everydocumentofacollectiontoselectthosedocumentsthatmatchthequerystatementThisscanishighlyinefficient andrequireMongoDBtoprocessalargevolumeofdata.

Indexesarespecialdatastructures,thatstoreasmallportionofthedatasetinaneasy-to-traverseform.The index stores the value of a specific field or set of fields, ordered by the value of the field asspecifiedintheindex.

TheensureIndex()Method

TocreateanindexyouneedtouseensureIndex()methodofMongoDB.ThebasicsyntaxofensureIndex( )methodis asfollows().

```
>db.COLLECTION_NAME.ensureIndex({KEY:1})
```

Herekeyisthenameofthefileonwhichyouwanttocreateindexand1isforascendingorder.Tocreateindexin descendingorderyouneed to use-1.

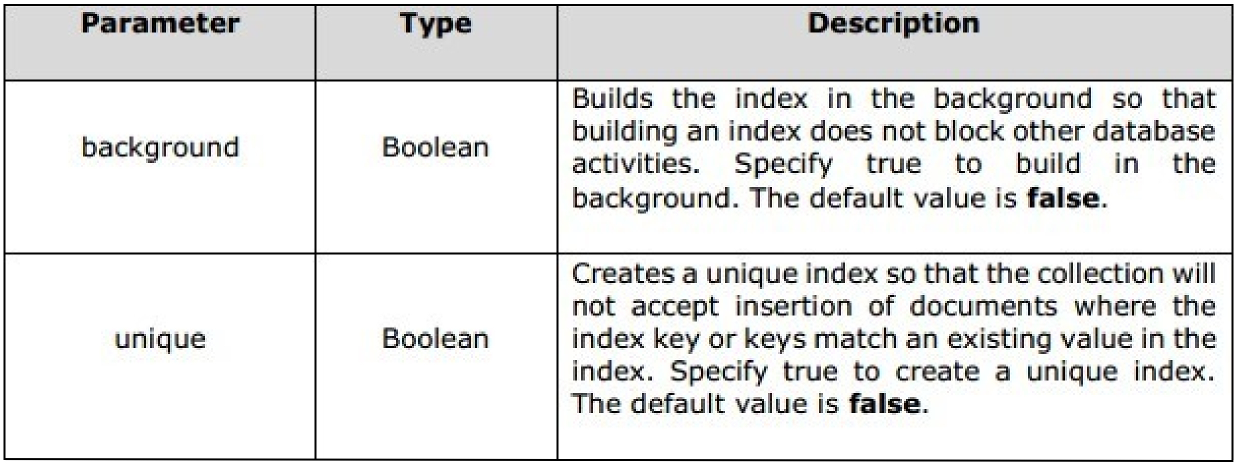
Example

```
>db.mycol.ensureIndex({"title":1})
```

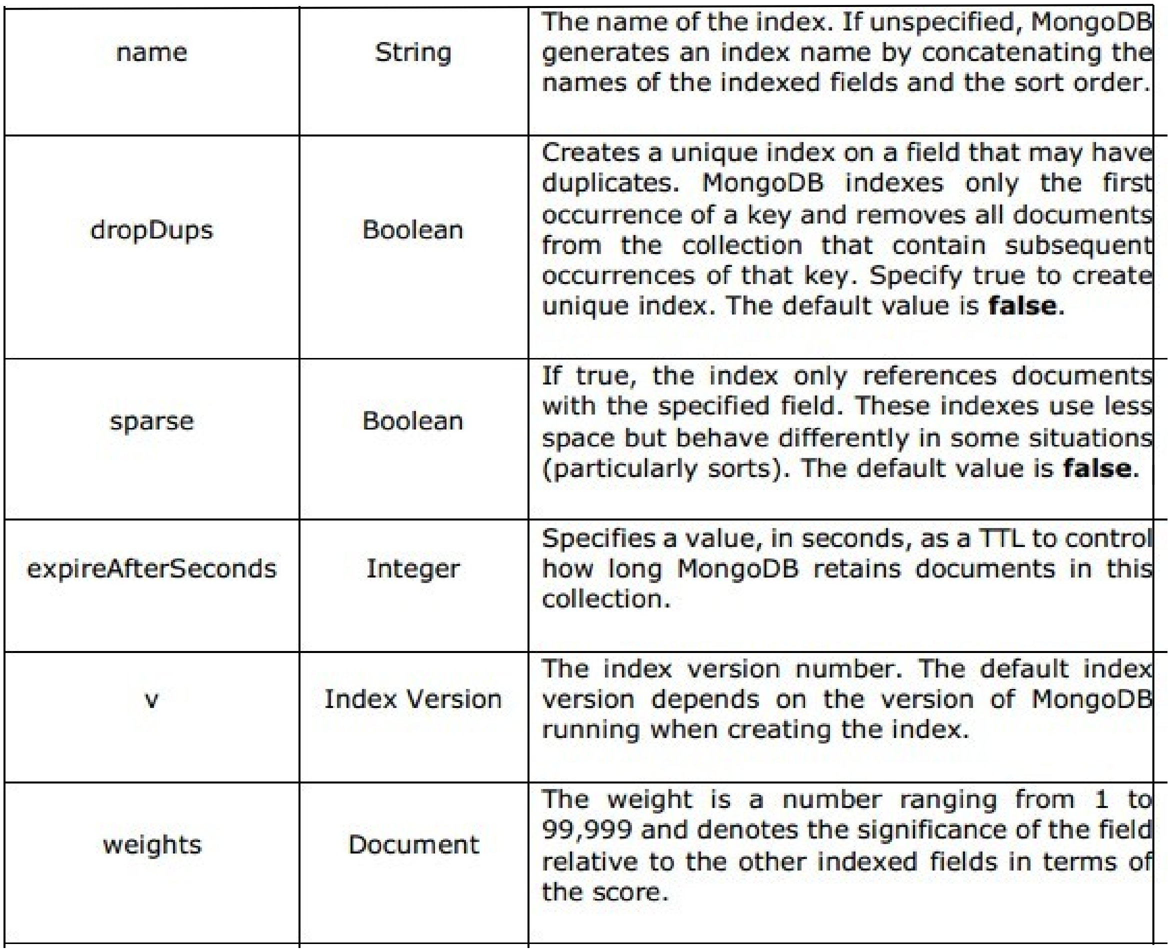
In**ensureIndex()**methodyoucanpassmultiplefields,tocreateindexonmultiplefields.

```
>db.mycol.ensureIndex({"title":1,"description":-1})
```

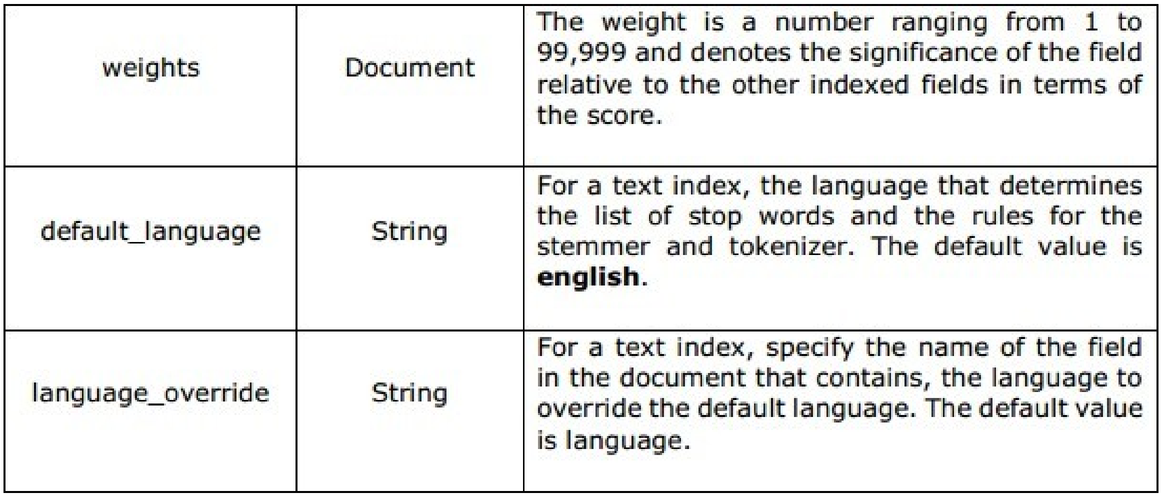
ensureIndex()methodalsoacceptslistofoptions(whichareoptional).Followingisthelist:



Parameter	Type	Description
background	Boolean	Builds the index in the background so that building an index does not block other database activities. Specify true to build in the background. The default value is **false**.
unique	Boolean	Creates a unique index so that the collection will not accept insertion of documents where the index key or keys match an existing value in the index. Specify true to create a unique index. The default value is **false**.



name	String	The name of the index. If unspecified, MongoDB generates an index name by concatenating the names of the indexed fields and the sort order.
dropDups	Boolean	Creates a unique index on a field that may have duplicates. MongoDB indexes only the first occurrence of a key and removes all documents from the collection that contain subsequent occurrences of that key. Specify true to create unique index. The default value is **false**.
sparse	Boolean	If true, the index only references documents with the specified field. These indexes use less space but behave differently in some situations (particularly sorts). The default value is **false**.
expireAfterSeconds	Integer	Specifies a value, in seconds, as a TTL to control how long MongoDB retains documents in this collection.
v	Index Version	The index version number. The default index version depends on the version of MongoDB running when creating the index.
weights	Document	The weight is a number ranging from 1 to 99,999 and denotes the significance of the field relative to the other indexed fields in terms of the score.



weights	Document	The weight is a number ranging from 1 to 99,999 and denotes the significance of the field relative to the other indexed fields in terms of the score.
default_language	String	For a text index, the language that determines the list of stop words and the rules for the stemmer and tokenizer. The default value is **english**.
language_override	String	For a text index, specify the name of the field in the document that contains, the language to override the default language. The default value is language.

Conclusion: -*Thus we havestudieduse andimplementationofaggregationfunction&indexingfunction.*

FAQ:-

1. Enlistvariousaggregationoperations.
2. ExplainMINfunctionwithexample.
3. ExplainPUSHfunction withexample.
4. Explain SUM&AVG function with example.

AssignmentNo.	**11**
Title	**MongoDB Map-reducesoperations:** ImplementMapreducesoperationwithsuitableexample usingMongoDB
PROBLEM STATEMENT/DEFINITION	ImplementMapreducesoperationwithsuitableexampleusingMongoDB
Objectives	• To understand concept of Map-reduce as data processing paradigm for condensing large volumes of data into useful *aggregated* results.
Software packages and hardware apparatus used	Operating Systems (64-Bit)64-BIT Fedora 17 or latest 64-BIT Update of Equivalent Open source OS or latest 64-BIT Version and update of Microsoft Windows 7 Operating System onwards Programming Tools (64-Bit) Latest Open source update of Eclipse Programming frame work, TC++, GTK++, mongoDB 2.6.
References	http://docs.mongodb.org/manual • MongoDB: The Definitive Guide, 2nd Edition,Powerful and Scalable Data Storage By Kristina Chodorow Publisher: O'Reilly Media
STEPS	Refer to details
Instructions for writing journal	1. Title 2. Problem Definition 3. Learning Objectives 4. Theory 5. Class Diagram/ER Diagram 6. Test cases 7. Program Listing 8. Output 9. Conclusion

AssignmentNo.11

Aim : ImplementMapreducesoperationwithsuitableexampleusingMongoDB

Objectives : To understand concept of Map-reduce as data processing paradigm for condensing large volumes of data into useful *aggregated* results

Theory : AspertheMongoDBdocumentation,MapReduceisadataprocessingparadigmforcondensinglargevolumes ofdataintousefulaggregatedresults.MongoDBusesmapReducecommandformap-reduceoperations.MapReduceis generallyusedforprocessinglargedatasets.

MapReduceCommand

Followingisthesyntaxofthebasicmapreducecommand

```
>db.collection.mapReduce(
        function(){emit(key,value);},//
        mapfunctionfunction(key,values)
        {returnreduceFunction},
        {//reducefunction
                out:
                collection,query:
                document,sort:
                document,limit:
                number
        } )
```

The map-reduce function first queries the collection, then maps the result documents toemitkey-valuepairs,whichisthenreducedbasedonthekeysthathavemultiplevalues.Intheabovesyntax –

- mapisajavascriptfunctionthatmapsavaluewithakeyandemitsakey-valuepair
- reduceisajavascriptfunctionthatreducesorgroupsallthedocumentshavingthesamekey
- outspecifiesthelocationofthemap-reducequeryresult
- queryspecifiestheoptionalselectioncriteriaforselectingdocuments
- sortspecifiestheoptionalsortcriteria
- limit specifies the optional maximum number of documents to be returned Using MapReduceConsiderthefollowingdocumentstructurestoringuserposts.Thedocumentstoresuser_nameofthe userand thestatus ofpost.

```
        {"post_text":"tutorialspointisanawesomewebsitefortutorials","use
                r_name": "mark",
                "status":"active"}
```

WewilluseamapReducefunctiononourpostscollectiontoselectalltheactiveposts,groupthemonthebasisofuser_nameandthen count thenumberofpostsbyeachuserusingthefollowingcode

```
>db.posts.mapReduce(

        function(){emit(this.user_id,1);},

        function(key, values) {return Array.sum(values)},

        {query:{status:"active"},
out:"post_total"})
```

TheabovemapReducequeryoutputs thefollowingresult −

```
{
"result" :
"post_total","timeMillis":
9,"counts":
{
"input": 4,
"emit":4,
"reduce":2,
"output": 2
},
"ok": 1,
}
```

The result shows that a total of 4 documents matched the query (status:"active"), themapfunctionemitted4documentswithkey-valuepairsandfinallythereducefunction

groupedmappeddocumentshavingthesamekeysinto2.

ToseetheresultofthismapReducequery,usethefindoperator−

>db.posts.mapReduce(function(){emit(this.user_id,1);},function(key,values){returnArray.sum(values)},
{query:{status:"active"},out:"post_total"}).find()

Theabovequerygivesthefollowingresult whichindicatesthatbothuserstomandmark havetwoposts in activestates−

```
{"_id": "tom","value":2}

{"_id":"mark","value":2}
```

Inasimilar manner, MapReduce queries can be used to construct large complex aggregation queries.Theuseofcustom JavascriptfunctionsmakeuseofMapReducewhichisveryflexibleandpowerful.

Conclusion:*ThuswehavestudiedMapreducefunction.*

FAQ:-

1. DefineandExplainmapreduceinMongoDBwithexamples.
2. WhytouseMapreduceinMongoDB
3. Explainthestructure ofObjectIDin MongoDB.
4. What are NoSQLdatabases?What are the different types of NoSQLdatabases?

AssignmentNo.	**12**
Title	**DatabaseConnectivity:**
PROBLEM STATEMENT/DEFINITION	WriteaprogramtoimplementMongoDBdatabaseconnectivitywithany frontendlanguagetoimplementDatabase navigationoperations(add,delete,editetc.)
Objectives	• Understand the concept of Connectivity between Java and databases • Understand how Java can invoke CRUD operation.
Software packages and hardware apparatus used	Operating Systems (64-Bit)64-BIT Fedora 17 or latest 64-BIT Update of Equivalent Open source OS or latest 64-BIT Version and update of Microsoft Windows 7 Operating System onwards Programming Tools (64-Bit) Latest Open source update of Eclipse Programming frame work, TC++, GTK++, mongoDB 2.6.
References	http://docs.mongodb.org/manual • MongoDB: The Definitive Guide, 2nd Edition,Powerful and Scalable Data Storage By Kristina Chodorow Publisher: O'Reilly Media
STEPS	Refer to details
Instructions for writing journal	1. Title 2. Problem Definition 3. Learning Objectives 4. Theory 5. Class Diagram/ER Diagram 6. Test cases 7. Program Listing 8. Output 9. Conclusion

AssignmentNo.12

Mini Project.	**13Group C Mini Project :**
Title	Using the **database concepts covered in Group A and Group B**, develop an application withfollowingdetails: 4. FollowthesameproblemstatementdecidedinAssignment-1ofGroupA. 5. FollowtheSoftwareDevelopmentLifecycleandotherconceptslearntin**SoftwareEngineeringCourse**throughouttheimplementation. 6. Developapplicationconsidering: • FrontEnd:Java/Perl/PHP/Python/ Ruby/.net/anyotherlanguage • Backend:MongoDB/MySQL/Oracle
PROBLEM STATEMENT/DEFINITION	
Objectives	
Software packages and hardware apparatus used	
References	
STEPS	
Instructions for writing journal	