# SHREE SWAMINARAYAN GURUKUL CAMPUS OPP.TRAFFIC POLICE STATION, SECTOR-22, GANDHINAGAR

**Contact No: 079-23245821 Website: sspolytechnic.co.in**



**DATABASE MANAGEMENT**

**(4331603)**

|  |  |
| --- | --- |
| **Student Name** |  |
| **Enrollment No** |  |
| **Semester** |  |
| **Branch** |  |
| **Batch** |  |

**INFORMATION TECHNOLOGY DEPARTMENT SHREE SWAMINARAYAN POLYTECHNIC SECTOR-22, GANDHINAGAR**

**SECTOR-22, GANDHINAGAR- 382022**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr No** | **Date** | **Experiment** | **Grade** | **Signature** |
| 1 |  | Prepare a report on current database trends, architecture and tools. |  |  |
| 2 |  | Draw ER-Diagram for Banking Management System and convert it  relational schema |  |  |
| 3 |  | Design the below given schemas using SQL  Command -”Create”. Decide the appropriate data type for each column. |  |  |
| 4 |  | Insert data in above tables using SQL Command “Insert”. |  |  |
| 5 |  | Retrieve data from tables in Practical 2 using Data SQL command- “Select” |  |  |
| 6 |  | Write SQL queries to use Update, alter, rename, delete, truncate and distinct. |  |  |
| 7 |  | Write SQL queries to use various date functions. |  |  |
| 8 |  | Write SQL queries to use various numeric functions |  |  |
| 9 |  | Write SQL queries to use various character functions. |  |  |
| 10 |  | Write SQL queries to use various conversion functions |  |  |
| 11 |  | Write SQL queries to use various group function and operators using tables created in Practical 1. |  |  |
| 12 |  | Write SQL query for set operators and join operations. |  |  |
| 13 |  | Apply the concept of integrity/data constraints while creating/altering a table: |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 14 |  | Write SQL queries for CREATE USER, GRANT, REVOKE AND DROP command |  |  |
| 15 |  | Write a program in PL/SQL to show the user of Cursor |  |  |
| 16 |  | Write a program in PL/SQL to show the user of Stored Procedures |  |  |
| 17 |  | Write a program in PL/SQL to show the user of Stored Function |  |  |
| 18 |  | Write a program in PL/SQL to show the user of Database Triggers |  |  |
| 19 |  | Normalization of database |  |  |

Date of submission:

Faculty Sign:



# PRACTICAL: 01

AIM: Prepare a report on current database trends, architecture and tools.



# PRACTICAL: 02

AIM: Draw ER-Diagram for Banking Management System and convert it relational schema

**Basic concepts of E-R**

|  |  |  |
| --- | --- | --- |
|  | Rectangle | An entity set is shown |
|  | Diamond | Relationship among a number of entities (1:1, 1:M, M:M) |
|  | Lines | Which link attribute to entity’s set to relationship sets |
|  | Ellipse | Which Represent Attribute |
|  | Double Ellipse | Which represent maulti value attributes |
|  | Dashed Ellipse | Which denoted derived attribute |
|  | Double Line | Which attributes total party station of an entity in a relationship set |
|  | Double rectangle | Which represent weak entity set |



# PRACTICAL: 03

AIM: Design the below given schemas using SQL Command -”Create”. Decide the appropriate data type for each column.

# THEORY:

Data Type in SQL

|  |  |
| --- | --- |
| **Data Type** | **Description** |
| Char(Size) | This data type is used to store character strings values of fixed length.  Maximum length is 255 characters |
| Varchar(size) Varchar2(size) | Stores character strings of variable length. Maximum length 2000 characters.  No default size will be considered. |
| Date | This data type is used to represent date & time  The format of Date: DD-MM-YY |
| Number(P,S) | Floating point number  pprecision in it cab be up to 38 digits  sScale, number of digits to the right of the decimal point |
| Number(p) | Fixed point number with a scale of zero & a precision of P |

DDL & DML Commands

* Data Definition Language The data definition language is used to create an object alter the structure of an object and also drop already created object. The Data Definition Languages used for table definition can be classified into following:
  + Create table
  + Alter table
  + Truncate table
  + Drop table



* **create**

This command is used to **create a new table** in SQL. The user has to give information like table name, column names, and their datatypes.

Syntax –

CREATE TABLE table\_name (

column\_1 datatype, column\_2 datatype, column\_3 datatype,

....

);

Example

CREATE TABLE Student\_info (

College\_Idnumber(2), College\_namevarchar(30), Branch varchar(10)

);

1. **Create a table ACCOUNT with column account number, name, city, balance, loan taken.**



1. **Create a LOAN table with column loan number, account number, loan amount, interest rate, loan date, and remaining loan.**
2. **Create table INSTALLMENT with column loan number, installment number column account, installment date and amount.**



1. **Create table TRANSACTION with number, transaction date, amount, type of transaction, mode of payment.**
2. **Show the structure of above tables using “Describe” command.**



# PRACTICAL: 04

AIM: Insert data in above tables using SQL Command “Insert”.

* + **insert**
    - After creation of table, it is necessary it should have data in it. The insert command is used to add data in form of one or more rows to a table as per follows

Syntax:

Insert into table\_name (column1, column2, ...) values (value1, value2, value3 ...);

Example:

INSERT INTO Student (ROLL\_NO, NAME, Age) VALUES ('5','PRATIK','19');

1 row created.

1. **Account**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **acc\_no** | **Name** | **City** | **Balance** | **Loan**  **\_taken** |
| **A001** | **Patel Jigar** | **Mehsana** | **50000** | **YES** |
| **A002** | **Patel Ramesh** | **Mehsana** | **50000** | **YES** |
| **A003** | **Dave Hardik** | **Ahmedabad** | **75000** | **NO** |
| **A004** | **SoniHetal** | **Ahmedabad** | **100000** | **NO** |
| **A005** | **SoniAtul** | **Vadodara** | **100000** | **YES** |



1. **Transaction**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Acc\_no** | **Tr\_date** | **Amt** | **Type\_of\_tr** | **Mode\_of\_pay** |
| **A001** | **1-may-20** | **10000** | **D** | **Cash** |
| **A002** | **3-july-20** | **5000** | **W** | **Cheque** |
| **A003** | **12-Aug-20** | **25000** | **D** | **Cheque** |
| **A004** | **15-may-20** | **30000** | **D** | **Cheque** |
| **A005** | **22-oct-20** | **15000** | **W** | **Cash** |



1. **Loan**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **loan\_ no** | **acc\_no** | **loan\_amt** | **Interest**  **\_rate** | **loan\_date** | **remaining\_ loan** |
| **L001** | **A001** | **100000** | **7** | **1-jan-20** | **75000** |
| **L002** | **A002** | **300000** | **9** | **18-may-20** | **150000** |
| **L003** | **A005** | **500000** | **11** | **15-june-20** | **300000** |



1. **Installment**

|  |  |  |  |
| --- | --- | --- | --- |
| **Loan\_no** | **Inst\_no** | **Date** | **Amount** |
| **L001** | **I001** | **2-Feb-04** | **15000** |
| **L002** | **I002** | **18-June-04** | **20000** |
| **L003** | **I003** | **15-July-04** | **20000** |



# PRACTICAL: 05

AIM: Retrieve data from tables in Practical 2 using Data SQL command- “Select”

# THEORY:

**SELECT COMMAND**

Stored information can be retrieved from the table through select command. Select is the mostfrequently used command, as access to information is needed all the time.

Syntax:

Select \* FROM TABLE NAME;

# EXAMPLE:

SQL> select \* from emp\_master;

The following command will select all the rows and columns from emp\_master.TABLE NAME: emp \_master

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EMPNO** | **ENAME** | **JOB** | **HIREDATE** | **SALARY** | **DEPT NO** |
| 1183 | STEVE | MANAGER | 11-5-2001 | 20000 | 10 |
| 1186 | MARK | CLERK | 22-1-1988 | 15000 | 10 |
| 1163 | MARTIN | HEAD CLERK | 1-1-1987 | 18000 | 20 |
| 2022 | SMITH | MANAGER | 24-9-2003 | 20000 | 20 |

Output:

4 rows selected.

Another Function

**Syntax:**

1. SELECT \* FROM (TableName);
2. SELECT (ColumnName1,ColumnName1) From (TableName)
3. SELECT \* FROM (TableName) WHERE (Condition);
4. Etc

**Example:**

1. SELECT \* FROM client\_master;
2. SELECT name, city FROM client\_master;
3. SELECT \* FROM client\_master Where city=’Bombay’;



* 1. **Display all rows and all columns of table Transaction.**



* 1. **Display all rows and selected columns of table Installment.**
  2. **Display selected rows and selected columns of table Account.**



* 1. **Display selected rows and all columns of table loan.**
  2. **Display the branch wise balance from account table.**



* 1. **Display list of those branches that have balance greater than 1 Lakh rupees.**
  2. **Display the list of customers in descending order of their name from account table.**



* 1. **Display those records where mode of payment is “cheque”**



# PRACTICAL: 06

AIM: Write SQL queries to use Update, alter, rename, delete, truncate and distinct.

Theory

* + - **Update**
      * The UPDATE statement in [SQL](https://www.geeksforgeeks.org/sql-tutorial/) is used to update the data of an existing table in the database.
      * We can update single columns as well as multiple columns using the UPDATE statement as per our requirement.

Syntax:

UPDATE *table\_name*

SET *column1* = *value1*, *column2* = *value2*, ... WHERE *condition*;

Example:

UPDATE Customers

SET ContactName = 'Amit', City= 'Ahmedabad' WHERE CustomerID = 1;

* **alter**

This command is used to **add, delete or change columns in the existing table**. The user needs to know the existing table name and can do add, delete or modify tasks easily.

Syntax

ALTER TABLE table\_name column\_name datatype;

Example –

In our Student\_info table, we want to add a new column for CGPA. The syntax would be as below as follows.

ALTER TABLE Student\_info ADD CGPA number;



* + - **truncate**

This command is used to **remove all rows from the table**, but the structure of the table still exists.

Syntax

remove an existing table.

TRUNCATE TABLE table\_name;

Example

The College Authority wants to remove the details of all students for new batches but wants to keep the table structure. The command they can use is as follows.

TRUNCATE TABLE Student\_info;

# RENAME:

It is possible to **change name of table with or without data in it using** simple RENAME command.

We can rename any table object at any point of time.

Syntax

RENAME <Table Name> To <New\_Table\_Name>;

Example:

If you want to change the name of the table from Employee to Emp we can use **rename** command as

RENAME TABLE Employee To EMP;

* + - **delete**

The DELETE statement is used to **delete existing records in a table**.

Syntax:

DELETE FROM table\_name WHERE condition;

Example:

DELETE FROM Customers WHERE CustomerName='Amit';



# DISTINCT:

**USE:**

* + - * To prevent the selection of distinct rows, we can include distinct clause with select command.
      * The following command will exclude duplicate empno.

Syntax:

Select distinct deptno from emp\_master;

Table: ACCOUNT.

1. **Change the name ‘pateljigar’ to ‘patelhiren’.**
2. **Change the name and city where account number is A005. (new name = ‘kotharinehal’and new city = ‘patan’).**



1. **Display only those records where loan taken status is ‘YES’.**
2. **Add the new column (address varchar2 (20)) into table ACCOUNT.**
3. **Create another table ACCOUNT\_TEMP (acc\_no, name, balance) from table ACCOUNT.**



1. **Rename the table ACCOUNT to ACCOUNT\_MASTER.**
2. **Delete the records whose account no is A004**



Table: LOAN.

1. **For each loan holders Add 100000 Rs. Amount into the column loan\_amt.**
2. **For each loan holders Increase the interest rate 2%.**
3. **Display only those records where loan holder taken a loan in month of January.**



1. **Modify the structure of table LOAN by adding one column credit\_no varchar2 (4).**
2. **Display the Loan amount\*2 of table LOAN.**
3. **Display the records of table LOAN by date wise in ascending order.**



1. **Display the records of table LOAN by account number wise in descending Order.**
2. **Increase the size 5 to 7 of column acc\_no.**



Table: INSTALLMENT.

1. **Change the Inst\_Date ‘2-Feb-04’ to ’3-Mar-04’.**
2. **Reduce 5000 amount from all Installment holders.**
3. **Add the amount 5000 where loan no is ‘L003’ and ‘L002’.**



1. **Change the column size of 5 to 7 where column name is Loan\_no.**
2. **Delete row where inst\_no is ‘I001’.**
3. **Only create a structure of table installment1 from table installment.**



Table: TRANSACTION.

1. **Insert any duplicate value and display all the records without any duplicate rows.**
2. **Select all the records in descending order(account number wise).**
3. **Display amt, date, and type of transaction by date wise**



1. **Delete a table TRANSACTION\_TEMP.**
2. **Display account number where type of transaction is ‘D**



# PRACTICAL: 07

AIM: Write SQL queries to use various date functions.

Theory:

This data type is used to store date and time

|  |  |
| --- | --- |
| **DATE** | format YYYY-MM-DD |
| **DATETIME** | format: YYYY-MM-DD HH:MI: SS |
| **TIMESTAMP** | format: YYYY-MM-DD HH:MI: SS |
| **YEAR** | format YYYY or YY |

1. **NOW()**
   * Returns the current date and time.

**SELECT NOW();**

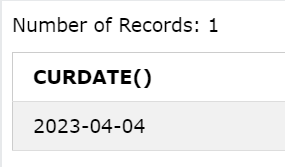
**Output:**



1. **CURDATE()**
   * Returns the current date.

**Query:**

SELECT CURDATE();

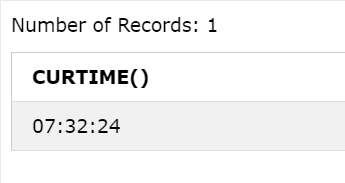


**CURTIME()**

* Returns the current time.

**Query:**

SELECT CURTIME();



Additional Example

* **ADD\_MONTHS()**

It returns the date after adding the number of months in specific date.

**Syntax:**

**Example:**

ADD\_MONTHS(date,number)

select ADD\_MONTHS (’04-jun-2009’,2) from dual;

Output: 04-AUG-2009

* **MONTHS\_BETWEEN**

It returns the number of months between the specified dates.

**Syntax:**

**Example:**

Output: 1

* **Round()**

MONTHS\_BETWEEN(date1,date2)

select MONTHS\_BETWEEN (’02-FEB-09’,’02-JAN-09’) from dual;

This function round a date in specified format

**Syntax:**

**Example:**

ROUND(date ,format)

select ROUND(sysdate, ’year’) from dual;

Output:-1-jan-2013



Exercise

1. **Add 3 months in current date.**



1. **Display months between date: ’17-02-04’ and ’17-02-05’.**
2. **Display last date of month ‘2-feb-06’.**



# PRACTICAL: 08

AIM: Write SQL queries to use various numeric functions

1. **Display integer value of 125.25.**
2. **Display absolute value of (-15)**
3. **Display floor value of 100.2**
4. **Display the square root of 16.**



1. **Display square root of 20.**
2. **Display result of 12 raised to 6.**
3. **Display result of 24 mod 2.**
4. **Show value of e3.**



1. **2 is angle in radius find out cos, sin, tan value of 2.**
2. **Show output of sign(-25), sign(25), sign(0).**



# PRACTICAL: 09

AIM: Write SQL queries to use various character functions.

1. **Find out length of string “hello world”.**
2. **Change the case of „HELLO WORLD‟ string to lower case.**
3. **Change the case of „hello world‟ to upper case.**



1. **Display each word initial letter as capital letter: government polytechnic for girls.**
2. **Find „put‟ from string „computer‟.**
3. **Add 10 star on left side of India string.**



1. **Add 10 star on right side of India string.**
2. **Trim „ion‟ from „information‟.**
3. **Trim „info‟ from „information‟.**



# PRACTICAL: 10

AIM: Write SQL queries to use various conversion functions

1. **Convert 100000 in format 99,99,99**
2. **Convert today‟s date to MM-DD-YY format**



# PRACTICAL: 11

AIM: Write SQL queries to use various group function and operators using tables created in Practical 1.

1. **Retrieve specified information for the account holder who are not in „Ahmedabad‟.**
2. **Retrieve specified information for the account holder who arenot in „Ahmedabad‟or**

„Vadodara‟.



1. **Retrieve those records of Account holder whose balance between is 50000 and 100000.**
2. **Retrieve those records of Account holder whose balance not between is 50000 and 100000.**



1. **Display only those records whose amount is 5000, 25000,30000.**
2. **Display only those records whose amount not in 5000, 25000, 30000.**



1. **Find minimum amount of transaction.**
2. **Find maximum amount of transaction.**
3. **Count the total account holders.**



1. **Count only that recordthat‟s made of payment is „cash‟.**
2. **Count only those records whose transaction made in the month of „MAY‟**
3. **Find the average value of transaction.**



1. **Display total balance for each branch from account table.**
2. **Display total balance for account in Ahmadabad city.**
3. **Find total amount of mode of pay „cash‟ from transaction table.**



# PRACTICAL: 12

AIM: Write SQL query for set operators and join operations.(Use tables of Practical 1)

1. **Display all customer account number who have account or taken loan from bank.**
2. **Display all customer account number who have account and also taken loan from bank.**



1. **Find the list of all account number of customer who have no loan in bank.**
2. **Display name of all customer whose remaining loan amount is greater than 50000.**



1. **Display account number of customer who have given installment of their loan on date 18-June-04.**
2. **Display name of all customer who have selected mode of payment as cash.**



# PRACTICAL: 13

AIM: Apply the concept of integrity/data constraints while creating/altering a table

1. Create table **SalesPeople** where Snumvarchar2(4) **P.K**(first letter should start with S) , Sname varchar2(20) **NOT NULL**,City Varchar2(15),Mobile\_No Number(10).



1. Create table Customer where cnum varchar2(4) **P.K**(first letter should start with C), Cname varchar2(20) **NOT NULL**,city varchar2(20),Rating number(3) **DEFAULT** 10,Snum number(4) **F.K** (where snum refers salespeople table)



1. Create table Order where

Order\_No number(4) **P.K**, Amount number(5), odate varchar2(10), cnum varchar2(4) **F.K**,(where cnum refers customer table), snum varchar2(4) **F.K** (where snum refers Salespeople table)



# PRACTICAL: 14

**AIM: Write SQL queries for CREATE USER, GRANT, REVOKE AND DROP command.**



# PRACTICAL: 15

AIM:Write a program in PL/SQL to show the user of Cursor



# PRACTICAL: 16

AIM:Write a program in PL/SQL to show the user of Stored Procedures



# PRACTICAL: 17

AIM:Write a program in PL/SQL to show the user of Stored Function.



# PRACTICAL: 18

AIM:Write a program in PL/SQL to show the user of Database Triggers



# PRACTICAL: 19

AIM: Normalization of database.

