**Database Management System**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Practical No : 6**    **Aim :** 1. Applying data constraints on tables   * Unique, primary key, foreign key, check, not null, default value     **Constraints** are the rules enforced on data columns or on table. These are used to limit the type of the data that can go into a table. This can ensure the accuracy and reliability of the data in the database.   * Table wise: Table-level constraints refer to one or more columns in the table. Table-level constraints specify the names of the columns to which they apply. Table-level CHECK constraints can refer to 0 or more columns in the table. * Column wise: Column-level constraints refer to a single column in the table and do not specify a column name (except check constraints). They refer to the column that they follow.   Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement.  CREATE TABLE table\_name (     column1 datatype *constraint*,     column2 datatype *constraint*,     column3 datatype *constraint*,     .... );  The following constraints are commonly used in SQL:   1. [NOT NULL](https://www.w3schools.com/sql/sql_notnull.asp) - Ensures that a column cannot have a NULL value 2. [UNIQUE](https://www.w3schools.com/sql/sql_unique.asp) - Ensures that all values in a column are different 3. [PRIMARY KEY](https://www.w3schools.com/sql/sql_primarykey.asp) - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table 4. [FOREIGN KEY](https://www.w3schools.com/sql/sql_foreignkey.asp) - Uniquely identifies a row/record in another table 5. [CHECK](https://www.w3schools.com/sql/sql_check.asp) - Ensures that all values in a column satisfies a specific condition 6. [DEFAULT](https://www.w3schools.com/sql/sql_default.asp) - Sets a default value for a column when no value is specified 7. [INDEX](https://www.w3schools.com/sql/sql_create_index.asp) - Used to create and retrieve data from the database very quickly 8. **NOT NULL**   The following SQL ensures that the "ID", "LastName", and "FirstName" columns will NOT accept NULL values when the "Persons" table is created:  *CREATE TABLE Persons (     ID int NOT NULL,     LastName varchar(255) NOT NULL,     FirstName varchar(255) NOT NULL,     Age int );*  To create a NOT NULL constraint on the "Age" column when the "Persons" table is already created, use the following SQL:  ALTER TABLE Persons MODIFY Age int NOT NULL;   1. **UNIQUE**   The following SQL creates a UNIQUE constraint on the "ID" column when the "Persons" table is created:  *CREATE TABLE Persons (     ID int UNIQUE,     LastName varchar(255) NOT NULL,     FirstName varchar(255),     Age int  );*  ALTER TABLE Persons ADD UNIQUE (ID);  ALTER TABLE Persons ADD CONSTRAINT ucpid UNIQUE (ID);{find all: user\_constraints}  ALTER TABLE Persons DROP CONSTRAINT ucpid;   1. **PRIMARY KEY**   The following SQL creates a PRIMARY KEY on the "ID" column when the "Persons" table is created:  CREATE TABLE Persons (     ID int PRIMARY KEY,     LastName varchar(255),     FirstName varchar(255),     Age int );  **On multiple column:**  CREATE TABLE Persons (     ID int,     LastName varchar(255),     FirstName varchar(255),     Age int,      CONSTRAINT PK\_Person PRIMARY KEY (ID,LastName)  );  ALTER TABLE Persons ADD PRIMARY KEY (ID);  ALTER TABLE Persons ADD CONSTRAINT PK\_Person PRIMARY KEY (ID,LastName);  ALTER TABLE Persons DROP CONSTRAINT PK\_Person;   1. **FOREIGN KEY**   The following SQL creates a FOREIGN KEY on the "PersonID" column when the "Orders" table is created:  CREATE TABLE Orders (     OrderID int  PRIMARY KEY,     OrderNumber int,     PersonID int FOREIGN KEY REFERENCES Persons(PersonID) );  or  CREATE TABLE Orders (     OrderID int,     OrderNumber int,     PersonID int,     PRIMARY KEY (OrderID),  CONSTRAINT FK\_PersonOrder FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)  );  ALTER TABLE Orders  ADD CONSTRAINT FK\_PersonOrder  FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);  ALTER TABLE Orders  DROP CONSTRAINT FK\_PersonOrder;   1. **CHECK**   The following SQL creates a CHECK constraint on the "Age" column when the "Persons" table is created. The CHECK constraint ensures that the age of a person must be 18, or older:  CREATE TABLE Persons (     ID int,     LastName varchar(255),     FirstName varchar(255),     Age int CHECK (Age>=18) );    CREATE TABLE Persons (     ID int NOT NULL,     LastName varchar(255) NOT NULL,     FirstName varchar(255),     Age int,     City varchar(255),      CONSTRAINT CHK\_Person CHECK (Age>=18 AND City='Sandnes')  );  ALTER TABLE Persons  ADD CONSTRAINT CHK\_PersonAge CHECK (Age>=18 AND City='Sandnes');  ALTER TABLE Persons DROP CONSTRAINT CHK\_PersonAge;   1. **DEFAULT**   The following SQL sets a DEFAULT value for the "City" column when the "Persons" table is created:  CREATE TABLE Persons (     ID int NOT NULL,     LastName varchar(255) NOT NULL,     FirstName varchar(255),     Age int,     City varchar(255) DEFAULT 'Sandnes' );  CREATE TABLE Orders (     ID int NOT NULL,     OrderNumber int NOT NULL,     OrderDate date DEFAULT GETDATE() );  ALTER TABLE Persons MODIFY City DEFAULT 'Sandnes';  ALTER TABLE Persons ALTER COLUMN City DROP DEFAULT;   1. INDEX : Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.   CREATE INDEX idx\_lastname ON Persons (LastName);  CREATE INDEX idx\_pname ON Persons (LastName, FirstName);  **1. Create the constraints as specified for all the tables.**    **a) Create table client (used to store client information) having following attributes:**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Column Name** | **Data Type** | **Size** | **Default** | **Attributes** | | clientno | varchar2 | 6 |  | Primary key at column level/first letter must start with ‘C’ (at column level) | | name | varchar2 | 20 |  | Not Null | | city | varchar2 | 15 |  |  | | pincode | number | 8 |  |  | | state | varchar2 | 15 |  |  | | baldue | number | 10, 5 |  |  |   **1. b) Create table product (used to store product information) having following attributes:**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Column Name** | **Data Type** | **Size** | **Default** | **Attributes** | | productno | varchar2 | 6 |  | Primary key at table level/first letter must start with ‘P’ (at table level) | | description | varchar2 | 15 |  | Not Null | | profitpercent | number | 4, 2 |  | Not Null | | unitmeasure | varchar2 | 10 |  | Not Null | | qtyonhand | number | 8 |  | Not Null | | reorderlvl | number | 8 |  | Not Null | | sellprice | number | 8, 2 |  | Not Null, Cannot be 0 | | costprice | number | 8, 2 |  | Not Null, Cannot be 0 |   **1. c) Create table salesman (used to store salesman information working for the company) having following attributes:**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Column Name** | **Data Type** | **Size** | **Default** | **Attributes** | | salesmanno | varchar2 | 6 |  | Primary key at table level /first letter must start with ‘S’ | | salesmanname | varchar2 | 20 |  | Not Null | | address1 | varchar2 | 30 |  | Not Null | | address2 | varchar2 | 30 |  |  | | city | varchar2 | 20 |  |  | | pincode | number | 8 |  |  | | state | varchar2 | 20 |  |  | | salamt | number | 8, 2 |  | Not Null, Cannot be 0 | | tgttoget | number | 6, 2 |  | Not Null, Cannot be 0 | | ytdsales | number | 6, 2 |  | Not Null | | remarks | varchar2 | 60 |  |  |   **1. d)** **Create table sales\_order (used to store client’s orders) having following attributes:**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Column Name** | **Data Type** | **Size** | **Default** | **Attributes** | | orderno | varchar2 | 6 |  | Primary key/first letter must start with ‘O’ | | clientno | varchar2 | 6 |  | Foreign key (at column level) references clientno of client table. | | orderdate | date |  |  | Not Null | | salesmanno | varchar2 | 6 |  | Foreign key (at column level) references salesmanno of salesman table. | | delaytype | char | 1 | F | Delivery : part (P) / full (F) | | billyn | char | 1 |  |  | | delaydate | date |  |  | Cannot be less than orderdate (at table level) | | orderstatus | varchar2 | 10 |  | Values(‘In Process’,’Fulfilled’,’BackOrder’,’Cancelled’) (at column level) |   **1. e)** **Create table sales\_order\_details (used to store client’s orders with details of each product ordered ) having following attributes:**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Column Name** | **Data Type** | **Size** | **Default** | **Attributes** | | orderno | varchar2 | 6 |  | Foreign key references (at table level) orderno of sales\_order table | | productno | varchar2 | 6 |  | Foreign key (at table level) references productno of product table | | qtyordered | number | 8 |  |  | | productrate | number | 10, 2 |  |  |   **2.** Alter table salesman, add constraint Not Null on remarks column and observe the behavior. Mention your remarks.  **3.** Insert data in all the tables as per Practical – 2 and check if any constraint is getting violated.  **4.** Delete data of salesman ‘S01’ from salesman table and observe the error. Rewrite the query for alteration of table, so that on deletion of ‘S01’ from salesman, corresponding values associated should also get deleted.  **5.** Delete data of order ‘O19001’ from sales\_order table and observe the error. Rewrite the query for alteration of table, so that if you remove ‘O19001’ from sales\_order, corresponding values associated should be set to NULL.  **6.** Drop primary key constraint on ‘orderno’ from sales\_order table. Observe the error. Write the drop query, so that associated constraints with ‘orderno’ also gets dropped. Check whether the constraints have dropped from user\_constraints table. |
|  |
|  |