**-----------------------------------------------------------------------------------------------**

**SQL2 Database creation,**

 **data types,**

 **DDL/DML Commands,**

 **constraints,**

 **table design with Constraints**

 **database relation ships**

**------------------------------------------------------------------------------------------------**

sql- structured query language

why sql ?

To work with data

**DATA TYPES**

**microsoft**

**kunal**

**arman**

**1,27,67**

 **. characters(letters) (David,shanthi)**

**(1)unicode(globalization) (1)Non-unicode(localization)**

 **(It supports all languages) (it supports only us eng)**

**(2)1 char - 2byte (2)1 char - 1 byte**

**Unicode Data types Non-unicode Data types**

**1)nchar(4000) 1)char(8000)(fixed length of characters)**

**2)nvarchar(4000) 2)varchar(17) yes, No, NotAplicabl**

**3)nvarchar(max)<1gb 3)varchar(max)<2gb**

**INT,CHAR,FLOAT,DECIMAL,DATE (DATA TYPES)**

**1 byte = 8 bits**

**1 char = 'A'**

**1 char = 'R'**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **1** | **0** | **1** | **1** | **0** | **1** | **0** |

**char vs varchar**

**1)use char for fixed, small sizes**

**2)use varchar to allow values of differing sizes**

**EmployeName (table column) char(7)**

**Kunal,Ram, David, JHON**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| D | A | V | I | D |  |  |

**jhon**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **J** | **H** | **O** | N |  |  |  |

**EmployeName (table column) varchar(7)**

**Armans**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **R** | **M** | **A** | **N** | **S** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **R** | **A** | **M** |

**RAM**

|  |  |  |
| --- | --- | --- |
| **R** | **A** | **M** |

**what is the difference between char, varchar and nvarchar datatypes?**

**char(n) :fixed length non-unicode character data type with length of n bytes.n must be a value from 1 through 8000**

 **It takes 1 byte per character. If we use char data type, there is memory wastage**

**varchar(n):Variable -length non-unicode character data type with length of nbytes. n must be a value from 1 through 8000**

 **It takes 1 byte per character**

 **If we use varchar data type, there is no memory wastage**

**nvarchar(n):Variable-length unicode character data type with length of n bytes. n must be a value from 1 through 4000**

 **It takes 2 bytes per character**

**Date & time :**

**1)date time(6bytes)**

**-----2)date time(7bytes) yyyy/mm/dd hr:mm:ss:ms**

**-----3)date 2014/06/13 yyyy/mm/dd**

**-----4)time hr:mm:ss:ms**

**EmpJoiningDate Date**

**Monetary Data:**

**1)small money(4bytes) lacks**

**2)money(8 bytes) crores**

**EmpSalary Money 10000.00/15000.00**

**Binary (binary means 1's & 0's)**

**1)binary data**

 **binary**

 **varbinary**

 **varbinary(max)**

**2)to store pictures, media, files etc**

**note: excel, xml supports unicode characters**

**CarImages varbinary**

**Numeric Data :**

**whole number**

 **tinyint(1byte)-->0 to 255**

 **smallint(2byte)**

 **int(4 bytes) 154667**

 **bigint(8 bytes)**

 **float(4 bytes) 145.34**

 **decimal(5,3) 35.635**

|  |  |  |
| --- | --- | --- |
| **Data type** | **Range** | **Storage** |
| Bigint | -**2^63** (-9,223,372,036,854,775,808) to **2^63-1** (9,223,372,036,854,775,807) | 8 Bytes |
| Int | -**2^31** (-2,147,483,648) to **2^31-1** (2,147,483,647) | 4 Bytes |
| smallint | -**2^15** (-32,768) to **2^15-1** (32,767) | 2 Bytes |
| Tinyint | **0 to 255** | 1 Byte |

**EmpId int**

**17891**

**17892**

**17893**

**17894**

**17895**

**EmpSalary decimal(5,2) 438.90**

**EmpSalary float**

**unique identifier**

**1)guid(16 byte)**

**`2)one per table**

**3)sql server insert new guid per each row using newid() function**

**Special data types:**

 **Time stamp(8 bytes)**

 **indicate activity**

 **known as 'Row version' column**

 **Changed whenever a row is modified**

 **not related to date and time**

 **updated by sql server engine itself**

 **only one per table**

**101,RAGAVA,HYD,2014/08/25,5000,UTYFR**

SQL COMMANDS

DDL- DATA DEFINATION LANGUAGE

 CREATE,ALTER,DROP,TRUNCATE

create table <table name>

(

 <col name> <data type> <primary key>,

 <col name> <data type> <null>,

 <col name> <data type> <not null>,

 <col name> <data type> <unique>,

 <col name> <data type> <null> <default(expression)>,

 <col name> <data type> <not null> check(expression),

 <col name> <data type> references PK\_TABLE(pk/uk)

)

**create table Employee**

 **( Empid int,**

 **,Ename varchar(20),**

 **,EmpLoc varchar(7),**

 **,EmpJoindate date,**

 **,EmpSalary float**

 **)**

**alter table Employee**

**add column EmpTax float**

**drop table Employee**

**truncate Employee**

DML- DATA MANIPULATION LANGUAGE

 INSERT,UPDATE,DELETE,SELECT

LOAD DATA INTO DATABASE TABLE

INSERT INTO EMP

VALUES(101,'RAM','ATLANTA', 2014/08/25, 5000)

MODIFY THE DATA IN DATABASE

UPDATE EMP

SET EmployeeLocation ='SEATTLE'

WHERE EmployeeId = 101

RETRIVE DATA FROM DATABASE

SELECT \* FROM EMP

load(INSERT) , modify(UPDATE), retrive(SELECT)

**Business Integrity(Constraints)**

**…………………………………………………….**

**Define constraints?**

**Three types of constraints:**

**1.domain : Not Null, Check, Default**

**2.entity : Primary key, Unique**

**3.referential : Foreign Key**

**Domain has the following constraints types:**

 **1.not null**

 **2.check**

 **3.default**

**Entity has the following constraints types**

 **1.primary key**

 **2.unique key**

**Referential has the following constraints types**

 **1.foreign key**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EmpId** | **EmpName** | **EmpSalary**  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Constraints**

 **not null**

 **check**

 **default**

 **primary key**

 **unique key**

 **foreign key**

**not null : to restrict null values in a colum**

 **ex: EmpName varchar(50) notnull**

**check :check constraint is used to check the values in a column with user defined condition**

 **ex: EmpAge int check(EmpAge>=25 and EmpAge<=50)**

 **Gender char(1) check(gender='m' or gender='f')**

**Default ex: EmpLocation varchar(15) Default ‘Atlanta’**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Empno INT Primary Key** | **EmpName varchar(17)** **NOT NULL** | **Gender char(1)****Check(gender=’M’ or gender=’F’)** | **DeptName varchar(17)****Defalult ‘SALES’** | **Salary int****Check(salary>=1000 and Salary<=5000)** |
| **151** | JHON | **M** | **Sales** | **2500** |
| **152** | **david** | **M** | **Sales** | **2000** |
| **153** | **Snehal** | **F** | **Sales** | **3000** |

 **If we create check constraint on single column i.e column level check constraint**

**check constraint expression will be involving more than one column i.e known as table level check constraint**

**default : to provide default value for a column**

 **The following ways in which we can provide default**

 **value in a table**

 **constant value ex: 0,1,n/a etc**

 **expression ex: price\*qty**

 **ex: EmpDept char(5) default 'SALES'**

**ex : JoiningDate date Default ‘28-05-2013’**

**What is the difference between Primary Key & Unique key**

**Primary key it wont allow duplicates and null values**

**Unique key it wont allow duplicates but null value**

**primary key : pk is used to identify a row. PK column does not allows duplicates and null values**

|  |  |  |
| --- | --- | --- |
| **Empid(Primary Key)** | **Passport(uk)** | **DLno(unique key)** |
| **1** | **123AB** | **Null** |
| **2** | **NULL** | **RTS675** |
| **3** | **NULL** | **RTS678** |
| **4** | **345B** | **Null** |

**candidate keys= empid,passport no, dlno**

**primary key = empid**

**alternate key = passport no, dlno**

**Ensures uniqueness**

**no null**

**auto creation of index(clustered)**

**one primary key per table**

**unique key : uk is used to identify a row. UK column does not allows duplicates but it allow null value**

**for alternate key**

**ensures uniqueness**

**allows null(only one null value allowed)**

**auto creation of index(non clustered)**

**many unique keys per table**

you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

**foreign key : fk is nothing but a column referencing values from other table primary key column**

**fk column allows duplicates and null values by default**

**parent child tables**

**FK in child table**

 **uk or pk in parent table**

**Department**

Create table Department ---master table

(

 DeptId int Primary key

 ,DeptName varchar(20) NOT NULL

)

**Department**

|  |  |
| --- | --- |
| **DeptId int Primary key** | **DeptName varchar(20) Not null**  |
| **10** | IT |
| **20** | **Financial** |
| **30** | **Sales** |

**Employee**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| EmpIdintPrimary key | EmpName Varchar(20)Not null | EmpSalAryVarchar(20)**Float** | EmpJoinigDate Date default ‘2014/01/31’ | EmpLoc Varchar(15) default ‘ATLANTA’ | EmpAge int Check(age>=20 and age<=40) | DrivingLic nvarchar(20)Unique key | DeptIDIntFk |
| 101 | Jhon | 13000.00 | 2014/01/31 | ATLANTA | 28 | Null | 10 |
| 102 | David | 20000.75 | 2014/01/31 | ATLANTA | 27 | 708AX | 10 |
| 103 | Arman | 27000 | 2014/01/31 | ATLANTA | 39 | 709SX | 20 |
| 104 | MAHANTH | 7000 | 2014/01/31 | ATLANTA | 40 | 8095T | 20 |
| 105 | RAM | 6000 | 2014/01/31 | ATLANTA | 20 | NULL | 20 |
| 106 | SRUJAN | 5000 | 2014/01/31 | ATLANTA |  25 | 908X | 10 |

**Create table Department**

**( Did int Primary Key,**

 **DeptName varchar(20) not null**

**)**

**Create table Employee**

**( Column name Datatype Constraint**

 **EmpId int Primary Key,**

 **EmpName varchar(20) Not Null,**

 **EmpSalary Float check(EmpSalary>=1000 and EmpSalary<5000**

 **EmpJoiningDate Date**

 **EmpLoc varchar(17) default ‘UK’**

 **DrivingLicence varchar(17) Unique**

 **Did int foreign key references Dept(Did) )**

**Does fk allows null values?**

**yes, fk allow null values**

**how many fk per table (many fk's)**

**Dept PARENT TABLE**

|  |  |
| --- | --- |
| **Deptid(pk)** | **DeptName** |
| **1** | **Accounts** |
| **2** | **Sales** |
| **3** | **Marketing** |

**Employee CHILD TABLE**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| EmpIdintPrimary key | EmpName Varchar(20)Not null | EmpSalAryFloat | EmpJoinigDate Date  | EmpLoc Date default ‘ATLANTA’ | EmpAge int Check(age>=20 and age<=40) | DrivingLic nvarchar(20)Unique key | DeptId intForeign Key |
| 101 | VENKAT | 13000 | 2014/01/31 | ATLANTA | 30 | Null | 1 |
| 102 | VASU | 20000 | 2014/01/31 | ATLANTA | 27 | 708AX | 2 |
| 103 | RAGAVA | 27000 | 2014/01/31 | ATLANTA | 39 | 709SX | 1 |
| 104 | HARISH | 7000 | 2014/01/31 | ATLANTA | 40 | 8095T | 1 |
| 105 | RAM | 6000 | 2014/01/31 | ATLANTA | 20 | 8095JK | 3 |
| 106 | SRUJAN | 5000 | 2014/01/31 | ATLANTA |  25 | 8095KLO | 3 |

[**http://www.w3resource.com/sql/joins/using-a-where-cluase-to-join-three-or-more-tables-based-on-a-parent-child-relationship.php**](http://www.w3resource.com/sql/joins/using-a-where-cluase-to-join-three-or-more-tables-based-on-a-parent-child-relationship.php)

--to view the structure of the table

SP\_HELP <TABLE NAME>

SP\_HELP Dept

To work with data

**create table Dept**

 **(**

 **Did int not null primary key**

 **DeptName varchar(7)**

 **)**

**create table Employee**

 **( Empid int not null primary key,**

 **,Ename varchar(20) unique not null**

 **,EmpTax int default '1000'**

 **,EmpJoindate date not null**

 **,EmpSalary money check(EmpSalary>0)**

 **,Deptid int Foreign key references Dept(Deptid)**

 **)**

 **How to add the Default Constraint to the already created table (Use following query to update it.**

**--Add a default constraint**

ALTER TABLE MyTable

ADD CONSTRAINT MyNewDefault

DEFAULT getdate() FOR MyNewCol2 ;

GO

**ALTER TABLE**

 **clients**

**ALTER COLUMN**

 **phone**

 **NVARCHAR(20) NOT NULL;**

**Database Relationships:**

relation ship is nothing but association or dependency among the tables

3 types of relation ships:

1.one to one

2.one to many

3.many to many

**1.one to one**

a row in one table associated with one row in another table is called one to one relationship

1-1 Relation ship :

**TICKET**

**CUSTOMER**

CID(PK) TNO(PK)

NAME SHOWTIME

ADDRESS DATE

CELLNO SCREENNO

EMAIL PRICE

 SEATNO

 CID(FK)

**1-1 Relation ship examples : 1 person will have 1 ssn/1 dL/ 1 Oter card**

**CUSTOMER**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CID(pk) | NAME | ADDRESS | CELLNO | EMAIL |
| 101 | Jhon | B.hils | 768-987-9988 | jhon@gmail |
| 102 | David | J.hils | 548-977-8978 | anil@gmail |
| 103 | Ragava | Madhapur | 678-907-4568 | ragva@gmail |

**TICKET**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tno(pk)** | **ShowTime** | **Date** | **ScreenNo** | **Price** | **SeatNo** | **Cid(FK)** |
| T1 | 8 am | 06/07/2011 | A1 | 100 | 23 A | 101 |
| T2 | 8 am | 06/07/2011 | A1 | 100 | 24 A | 102 |
| T3 | 8 am | 06/07/2011 | A1 | 100 | 25 A | 103 |

**example: create "Customer", "MovieTicket" tables and maintain one to one relation ship**

create table Customer

 (

 CID int Primary Key),

 NAME varchar(50),

 ADDRESS nvarchar(50),

 CELLNO int )

 create table Ticket

 (

 TNO int Primary Key,

 SHOWTIME time,

 DATE date,

 SCREENNO varchar(8),

 PRICE float,

 SEATNO varchar(8),

 CID int foreign key references Customer(CID)

 )

Examples :

* One person will have one ssn no

 **2.**

 **3.**

**2.one to many relationship:**

a row in one table associated with many rows in another table is called one to many relationship

**dept PARENT TABLE**

|  |  |
| --- | --- |
| **Deptno (pk)** | **Dname** |
| **10** | **IT** |
| **20** | **HR** |
| **30** | **SALES** |
| **40** | **FINANCE** |

**Emp CHILD TABLE 1 dept(pk) -- many employee (fk)**

|  |  |  |
| --- | --- | --- |
| **Empno(pk)** | **Ename** | **Deptno(fk)** |
| **1** | **Jhon** | **10** |
| **2** | **RAM TEJ** | **10** |
| **3** | **Reddy** | **10** |
| **4** | **HARISH** | **20** |
| **5** | **VENKAT** | **30** |

**1-M**

**1 Dept(pk) M emp(fk) pk**

**Examples :**

**1.One college department have many students**

**2. One department have many employees**

**3.**

**1 dept – M Employee**

**example: create dept,employee tables and maintain one to many relationship**

create table dept ---Parent table/Master table

 ( deptno int primary key

 ,dname varchar(15)

 ,dloc varchar(15)

 )

Create table Employee --child /transaction

(EmpId int Primary key

,EmpName varchar(20) not null

,EmpSalary float

,EmpJoiningDate Date

,EmpLoc varchar(20) default ‘ATLANTA’

,EmpAge int check(age>=20 and age<=40)

,DrivingLicNo nvarchar(20) unique key

,DeptId int Foreign key references Department(DeptId)

)

**3.many to many relationship**

many rows in one table associated with many rows in another table is called many to many relation ship

 it will be established with the help of linking table

 linking table contains multiple foreign keys and composite primary key

**author M M Books**

|  |  |
| --- | --- |
| **Auid(pk)** | **Auname** |
| **1** | **Balu** |
| **2** | **TataMcGrewHill** |
| **3** | **Bala** |
|  |  |
| **Bookid(pk)** | **Title** |
| **B1** | **C** |
| **B2** | **C++** |
| **B3** | **Java** |
| **B4** | **.net** |

**books**

**M-M 2PK,2FK**

**author books**

|  |  |  |  |
| --- | --- | --- | --- |
| **abid(pk)** | **Auid(f.k)** | **Bookid(f.k)** | **Cost**  |
| **171** | **1** | **B1** | **300** |
| **172** | **1** | **B2** | **200** |
| **173** | **2** | **B3** | **500** |
| **174** | **2** | **B4** | **700** |

**junction table (or) linking table**

**Note: Always we define the foreign key in child table**

 **1 Authour M Books**

 **1 Book M Authours**

**Emp**

**OterCard**

**parent table child table**

 **or or**

**master table detail table**

-->child table always depends on parent table

-->here OterCard table is the child table, it exists only emp table is there.

**what is the difference between pk & uk?**

**pk is used to identify a row. pk column does not allows duplicates and null values.clustered index is created**

**uk is used to identify a row. uk column does not allows duplicates**

**but it allow one null value. non clustered index is created**

**self referential table:**

|  |  |  |
| --- | --- | --- |
| **Empid(pk)** | **Name** | **Mid(fk)** |
| **101** | **Dheeru** | **null** |
| **102** | **Kokila** | **null** |
| **103** | **Mukesh** | **101** |
| **104** | **Anil** | **101** |
| **105** | **Shiva** | **103** |

**can we create fk using uk?**

yes we can create fk using pk & uk

**Cascaded features:**

**1) what if pk/uk column in parent table changes**

**2) what if a row in parent table is deleted**

**3) solution= cascaded update**

 **cascaded delete**

**cascade update**

 **on**

 **off(no action will be done)**

**cascade delete**

 **on**

 **off(no action will be done)**

**create table students**

 **( sid int primary key identity(1,1)**

 **,sname varchar(20) unique not null**

 **,sage int default 24**

 **,joindate date time not null**

 **,fees money check(fees>0)**

 **)**

**Identity property :**

**1)identity property can be set for**

 **tinyint, small int, int , bigint, decimal, numeric**

**2)only one column per table**

**3)no null & default values allowed**

**4)unique values**

**5)beware of gaps**

 **use SET IDENTITY\_INSERT ON to fill an existing gap**

 SET IDENTITY\_INSERT dbo.Person on

 insert into Person(id,name,location,number)

 values(8,'aaa','kdf',12345678)

 SET IDENTITY\_INSERT dbo.Person off

**identity will take 2parameters identity(1,1)**

**1)seed (stating value)**

**2)increament(increament value)**

 **IDENTITY(1,5)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Empid** | **Name** | **Gender** | **Grade** | **salary** |
| **1** | **A** | **M** | **A** | **4000** |
| **5** | **B** | **M** | **A** | **3000** |
| **10** | **C** | **F** | **B** | **2000** |
| **15** | **D** | **F** | **B** | **3000** |

ASSIGNMENTS /Tasks:

* Create a products & customer table which have

 ProductId,ProductName,ProductCost in Products Table

& customerid,customername,customerloc,customerEmailid,ProductId,CustomerCreditPoints, customerDrivingLicNo,ShippingDate,BillingDate columns in customer table

Add the appropriate data types

Business Rules: apply constraints

Productid should n’t allow duplicates & null values

ProductName should n’t allow null values

Customer id data should n’t allow duplicates & null values

CustomerName,CustomerLoc,Emailid should n’t allow null values

CustomerCreditsPoints should in between 20 &67

CustomerDrivingLicNo should n’t allow duplicate but allow null

Make a relation between products,customer table with column ProuctId

In customer table, ProductId is FK

ShippingDate for all customers on same date 2014-07-26 or use ur system date with the help Of getDate() function.

* Create employee & department table & make a relation between them
* Create your own example on 1-1,1-m,m-m Relationships