Microsoft SQL Server

3 Steps   View most recent version on Tango.us

Created by	Creation Date	Last Updated
Surojeet Dey	Aug 28, 2024	Aug 28, 2024

STEP 1

Creation of database and tables. The Codes are --

1 - Database - CREATE DATABASE Telecommunication_db

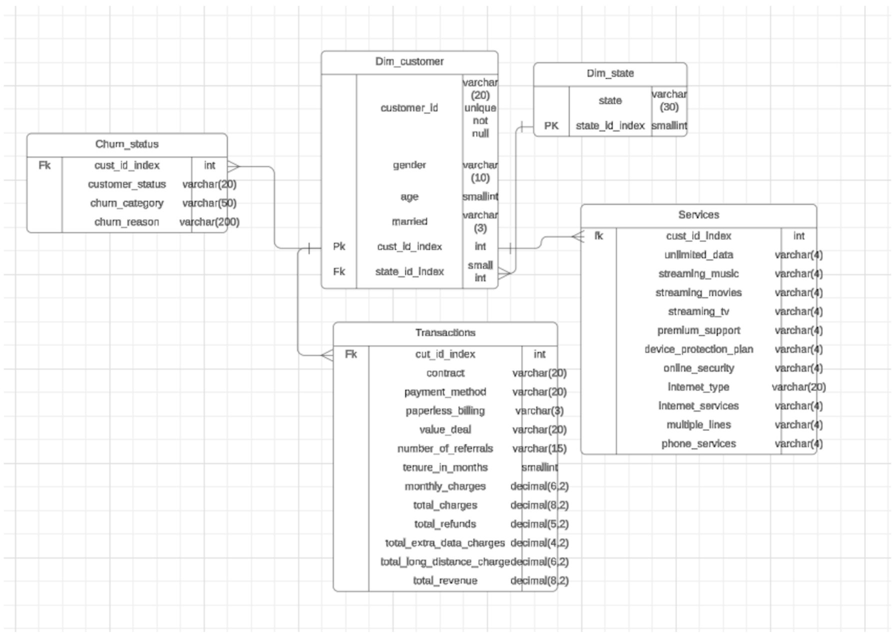
2 - Creation of Tables -

```
IF OBJECT_ID('Dim_state', 'U') IS NOT NULL DROP TABLE Dim_state;
IF OBJECT_ID('Dim_customer', 'U') IS NOT NULL DROP TABLE Dim_customer;
IF OBJECT_ID('Churn_status', 'U') IS NOT NULL DROP TABLE Churn_status;
IF OBJECT_ID('Services', 'U') IS NOT NULL DROP TABLE Services;
IF OBJECT_ID('Transactions', 'U') IS NOT NULL DROP TABLE Transactions;
-- Create Dim_state table
CREATE TABLE [Dim_state] (
[state] varchar(30),
[state_id_index] smallint PRIMARY KEY
);
-- Create Dim_customer table
CREATE TABLE [Dim_customer] (
[customer_id] varchar(20) UNIQUE NOT NULL,
[gender] varchar(10),
[age] smallint,
[married] varchar(3),
[cust_id_index] int PRIMARY KEY,
[state_id_index] smallint,
CONSTRAINT [FK_Dim_customer_state_id_index]
FOREIGN KEY ([state_id_index])
REFERENCES [Dim_state]([state_id_index])
);
-- Create indexes on Dim_customer table
CREATE INDEX [Idx_Dim_customer_cust_id_index]
ON [Dim_customer] ([cust_id_index]);
CREATE INDEX [Idx_Dim_customer_state_id_index]
ON [Dim_customer] ([state_id_index]);
-- Create Churn_status table
CREATE TABLE [Churn_status] (
[cust_id_index] int,
[customer_status] varchar(20),
[churn_category] varchar(50),
[churn_reason] varchar(200),
CONSTRAINT [FK_Churn_status_cust_id_index]
FOREIGN KEY ([cust_id_index])
REFERENCES [Dim_customer]([cust_id_index])
);
-- Create index on Churn_status table
CREATE INDEX [Idx_Churn_status_cust_id_index]
ON [Churn_status] ([cust_id_index]);
-- Create Services table
```

```
CREATE TABLE [Services] (
[cust_id_index] int,
[unlimited_data] varchar(4),
[streaming_music] varchar(4),
[streaming_movies] varchar(4),
[streaming_tv] varchar(4),
[premium_support] varchar(4),
[device_protection_plan] varchar(4),
[online_security] varchar(4),
[internet_type] varchar(20),
[internet_services] varchar(4),
[multiple_lines] varchar(4),
[phone_services] varchar(4),
CONSTRAINT [FK_Services_cust_id_index]
FOREIGN KEY ([cust_id_index])
REFERENCES [Dim_customer]([cust_id_index])
);
-- Create index on Services table
CREATE INDEX [Idx_Services_cust_id_index]
ON [Services] ([cust_id_index]);
-- Create Transactions table
CREATE TABLE [Transactions] (
[cust_id_index] int, -- Fixed typo from "cut_id_index"
[contract] varchar(20),
[payment_method] varchar(20),
[paperless_billing] varchar(3),
[value_deal] varchar(20),
[number_of_referrals] varchar(15),
[tenure_in_months] smallint,
[monthly_charges] decimal(6,2),
[total_charges] decimal(8,2),
[total_refunds] decimal(5,2),
[total_extra_data_charges] decimal(8,2),
[total_long_distance_charge] decimal(8,2),
[total_revenue] decimal(8,2),
CONSTRAINT [FK_Transactions_cust_id_index]
FOREIGN KEY ([cust_id_index])
REFERENCES [Dim_customer]([cust_id_index])
);
-- Create index on Transactions table
CREATE INDEX [Idx_Transactions_cust_id_index]
ON [Transactions] ([cust_id_index]);
```

STEP 2

Entity Relationship Diagram [ERD]


Dim_customer
customer_id varchar (20) unique not null
gender varchar (10)
age smallint
married varchar (3)
Pk cust_id_index int
Fk state_id_index small int
Dim_state
state varchar (30)
PK state_id_index smallint
Churn_status
Fk cust_id_index int
customer_status varchar(20)
churn_category varchar(50)
churn_reason varchar(200)
Services
fk cust_id_index int
unlimited_data varchar(4)
streaming_music varchar(4)
streaming_movies varchar(4)
streaming_tv varchar(4)
premium_support varchar(4)
device_protection_plan varchar(4)
online_security varchar(4)
Internet_type varchar(20)
Internet_services varchar(4)
multiple_lines varchar(4)
phone_services varchar(4)
Transactions
Fk cut_id_index int
contract varchar(20)
payment_method varchar(20)
paperless_billing varchar(3)
value_deal varchar(20)
number_of_referrals varchar(15)
tenure_in_months smallint
monthly_charges decimal(6,2)
total_charges decimal(8,2)
total_refunds decimal(5,2)
total_extra_data_charges decimal(4,2)
total_long_distance_charge decimal(6,2)
total_revenue decimal(8,2)

STEP 3

Exploratory Data Analysis on SQL

```
--ERD Exploratory data analysis
--1--Total_customer = 6418
select COUNT(*) as total_customer
from Dim_customer;

--2--New_joiner = 411
select COUNT(*) as new_jooiner
from Churn_status
where customer_status= 'Joined';

--3--total_churn = 1732
select count(*) as total_churn
from Churn_status
where customer_status = 'Churned';

--4--Churn_rate = 27.0%
select
ROUND(
100.0*sum(case when customer_status = 'Churned'then 1 else 0 end)/count(customer_status),1 )
from Churn_status;
```

```
select * from Dim_customer;
select * from Churn_status;

--4 --total_churn_by_gender
with
total_churn_by_gender as
(
select gender,
count(*) as churned_customer_cnt,
(select count(*) from Churn_status where customer_status = 'churned')as all_churn_cust
from Dim_customer as dc
inner join Churn_status as cs on dc.cust_id_index=cs.cust_id_index
where customer_status = 'churned'
group by gender
)
select gender, round(100.0*churned_customer_cnt/all_churn_cust,2)
from total_churn_by_gender;

--5-- churn_rate_by_payment_method
with
all_and_lost_customer as
(
select payment_method,COUNT(*) as start_customer,
sum(case when customer_status = 'Churned' then 1 else 0 end) as lost_customer
from Transactions as t
inner join Churn_status as c on t.cust_id_index=c.cust_id_index
group by payment_method
)
select payment_method,(100.0*lost_customer/start_customer)
from all_and_lost_customer;

--6--churn_rate_by_contract
with
initial_and_lost_contract as
(
select contract,COUNT(*) as initial_contract,
sum(case when customer_status='churned' then 1 else 0 end)as lost_contract
from Transactions as t
inner join Churn_status as c on t.cust_id_index=c.cust_id_index
group by contract
)
select contract, (100.0*lost_contract/initial_contract) as churn_rate_by_contract
from initial_and_lost_contract;

--7-- churn_rate_by_state
with
state_wise_all_and_lost_customer as
(
select state , count(*) as initial_customer_cnt,
sum(case when customer_status='churned' then 1 else 0 end) as lost_customer_cnt
from Dim_customer as c
inner join Dim_state as s on c.state_id_index=s.state_id_index
inner join Churn_status as cs on c.cust_id_index=cs.cust_id_index
group by state
)
select top 5 state, (100.0*lost_customer_cnt/initial_customer_cnt) as
churn_rate_by_state
from state_wise_all_and_lost_customer
order by churn_rate_by_state desc;

--8-- Total_churn_by_churn_category
select churn_category,count(*)as churn_cnt
from Churn_status
where churn_category <> 'not_available'
group by churn_category
order by churn_cnt desc ;

--9-- churn_rate_by_internet_type
with
initial_and_lost_customer_on_internet_type as
(
select internet_type, COUNT(*)as initial_customer_cnt,
sum(case when customer_status = 'churned' then 1 else 0 end) as lost_customer
from Churn_status as c
inner join Services as s on c.cust_id_index=s.cust_id_index
group by internet_type
)
select internet_type,(100.0*lost_customer/initial_customer_cnt)as churn_rate_by_inter-
net_type
from initial_and_lost_customer_on_internet_type;

--10-- total_customers_and_churn_rate_by_agegroup
with
age_group as
(
select cust_id_index,age,
case
WHEN age >= 0 AND age < 20 THEN '0-20'
WHEN age >= 20 AND age < 35 THEN '20-35'
WHEN age >= 35 AND age < 50 THEN '35-50'
WHEN age >= 50 THEN 'Above 50' end as age_group
from Dim_customer
),
customer_age_group_churn_info as
(
select age_group,COUNT(*) as total_customer_cnt,
sum(case when customer_status='churned' then 1 else 0 end) as churn_customer_cnt
from age_group as ag
inner join Churn_status as cs on ag.cust_id_index=cs.cust_id_index
group by age_group
)
select age_group, total_customer_cnt,
(100.0*churn_customer_cnt/total_customer_cnt) as churn_rate_by_customer
from customer_age_group_churn_info
order by age_group;
select * from Churn_status;
select * from Transactions;

--11-- total_customers_and churn rate by tenure_group
with
tenure_groups as
(
select distinct cust_id_index,
case
when tenure_in_months>=0 and tenure_in_months<6 then '0-6'
when tenure_in_months>=6 and tenure_in_months<12 then '6-12'
when tenure_in_months>=12 and tenure_in_months<18 then '12-18'
when tenure_in_months>=18 and tenure_in_months<24 then '18-24'
else 'Above 24'
end as Tenure_group
from Transactions
order by Tenure_group
),
tenure_group_initial_and_lost_cust as
(
select tenure_group,COUNT(*) as total_customer_cnt,
sum(case when customer_status='churned' then 1 else 0 end) as churn_customer_cnt
from tenure_groups as ag
inner join Churn_status as cs on ag.cust_id_index=cs.cust_id_index
group by tenure_group
)
select tenure_group, total_customer_cnt,
(100.0*churn_customer_cnt/total_customer_cnt) as churn_rate_by_customer
from tenure_group_initial_and_lost_cust
order by Tenure_group
;
select SUM(total_revenue)
from Transactions;
with
age_group as
(
select cust_id_index,age,
case
WHEN age >= 0 AND age < 20 THEN '0-20'
WHEN age >= 20 AND age < 35 THEN '20-35'
WHEN age >= 35 AND age < 50 THEN '35-50'
WHEN age >= 50 THEN 'Above 50' end as age_group
from Dim_customer
)
select gender,age_group, SUM(total_revenue)-SUM(total_refunds)
from Transactions as t
inner join Dim_customer as c on t.cust_id_index=c.cust_id_index
inner join age_group as a on a.cust_id_index=c.cust_id_index
group by gender,age_group
order by age_group
```