

SQL DEMONSTRATION
CODE FOR PRESENTATION:

-- Table: public.payment

-- DROP TABLE public.payment;

```
CREATE TABLE public.payment
(
    payment_id integer NOT NULL,
    payment_type text ,
    CONSTRAINT payment_pkey PRIMARY KEY (payment_id)
);
```

--file path to this payment csv: /Users/benjaminmcdaniel/Desktop/d211_churn /payment.csv

-- Table: public.location

-- DROP TABLE public.location;

```
CREATE TABLE public.location
(
    location_id integer NOT NULL,
    zip integer,
    city varchar(100),
    state varchar(2),
    county varchar(100),
    CONSTRAINT location_pkey PRIMARY KEY (location_id)
);
```

--file path to this location csv: /Users/benjaminmcdaniel/Desktop/d211_churn /location.csv

-- Table: public.job

-- DROP TABLE public.job;

```
CREATE TABLE public.job
(
    job_id integer NOT NULL,
    job_title varchar(100),
    CONSTRAINT job_pkey PRIMARY KEY (job_id)
);
```

--file path to job csv: /Users/benjaminmcdaniel/Desktop/d211_churn /job.csv

-- Table: public.contract

```
-- DROP TABLE public.contract;
```

```
CREATE TABLE public.contract
```

```
(  
    contract_id integer NOT NULL,  
    duration VARCHAR(30),  
    CONSTRAINT contract_pkey PRIMARY KEY (contract_id)  
);
```

```
--file path to contract csv: /Users/benjaminmcdaniel/Desktop/d211_churn /contract.csv
```

```
-- Table: public.customer
```

```
-- DROP TABLE public.customer;
```

```
CREATE TABLE public.customer
```

```
(  
    customer_id text NOT NULL,  
    lat numeric,  
    lng numeric,  
    population integer,  
    children integer,  
    age integer,  
    income numeric,  
    marital text ,  
    churn text ,  
    gender text ,  
    tenure numeric,  
    monthly_charge numeric,  
    bandwidth_gp_year numeric,  
    outage_sec_week numeric,  
    email integer,  
    contacts integer,  
    yearly_equip_faiure integer,  
    techie text,  
    port_modem text ,  
    tablet text ,  
    job_id integer,  
    payment_id integer,  
    contract_id integer,  
    location_id integer,  
    CONSTRAINT customer_pkey PRIMARY KEY (customer_id),  
    CONSTRAINT customer_contract_id_fkey FOREIGN KEY (contract_id)  
        REFERENCES public.contract (contract_id) MATCH SIMPLE  
    ON UPDATE NO ACTION
```

```

        ON DELETE NO ACTION
        NOT VALID,
CONSTRAINT customer_job_id_fkey FOREIGN KEY (job_id)
    REFERENCES public.job (job_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID,
CONSTRAINT customer_location_id_fkey FOREIGN KEY (location_id)
    REFERENCES public.location (location_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID,
CONSTRAINT customer_payment_id_fkey FOREIGN KEY (payment_id)
    REFERENCES public.payment (payment_id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
    NOT VALID
);
--file path to customer csv: /Users/benjaminmcdaniel/Desktop/d211_churn /customer.csv

-- new table for the add-on for services
create table services (
    customer_id varchar(30),
    internetservice varchar(25),
    phone varchar(3),
    multiple varchar(3),
    Onlinesecurity varchar(3),
    Onlinebackup varchar(3),
    DeviceProtection varchar(3),
    techsupport varchar(3),
    CONSTRAINT services_pkey PRIMARY KEY (customer_id)
);
--file path to services csv: /Users/benjaminmcdaniel/Desktop/d211_churn /Services_x.csv

create table survey_responses (
    customer_id varchar(30),
    timely_responses int,
    timely_fixes int,
    timely_replacement int,
    reliability int,
    "options" int,
    respectful int,
    courteous int,
    active_listening int,

```

```

        CONSTRAINT survey_responses_pkey PRIMARY KEY (customer_id)
    );
--file path to survey responses csv: /Users/benjaminmcdaniel/Desktop/d211_churn
/Survey_Responses.csv

--End of churn tables DDL
--Begin census table ddl

-- Table: public.census
-- DROP TABLE public.census
CREATE TABLE public.census
(
    index integer,
    sumlev integer,
    region integer,
    division integer,
    state_num integer,
    state_name varchar(30),
    population2019 integer,
    state_abbr varchar(2) NOT NULL,
    CONSTRAINT census_pkey PRIMARY KEY(index)
);
--file path to census csv: /Users/benjaminmcdaniel/Desktop/d211_churn /censusest2019.csv

--file path to data set table data
--/Users/benjaminmcdaniel/Desktop/d211_churn /payment.csv
--/Users/benjaminmcdaniel/Desktop/d211_churn /location.csv
--/Users/benjaminmcdaniel/Desktop/d211_churn /job.csv
--/Users/benjaminmcdaniel/Desktop/d211_churn /contract.csv
--/Users/benjaminmcdaniel/Desktop/d211_churn /customer.csv
--/Users/benjaminmcdaniel/Desktop/d211_churn /Services_x.csv
--/Users/benjaminmcdaniel/Desktop/d211_churn /Survey_Responses.csv
--/Users/benjaminmcdaniel/Desktop/d211_churn /censusest2019.csv

--telecom_master table creation from a query

CREATE TABLE telecom_master AS
SELECT c.customer_id, c.lat as latitude, c.lng as longitude,
       c.population as local_population, c.children, c.age,
       c.income, c.marital, c.churn, c.gender, c.tenure, c.monthly_charge,

```

```

        c.bandwidth_gp_year AS bandwidth_gb_year, c.outage_sec_week, c.email,
        c.contacts, c.yearly_equip_faiure AS yearly_equip_failure,
        c.techie, c.port_modem, c.tablet, p.payment_type,
        con.duration AS contract_type,
        l.zip, l.city, l.state, l.county,
        cen.region, cen.division, cen.state_num, cen.state_name,
        cen.population2019
FROM customer AS c
INNER JOIN
payment as p
ON c.payment_id = p.payment_id
INNER JOIN
contract as con
ON c.contract_id = con.contract_id
INNER JOIN
location as l
ON c.location_id = l.location_id
INNER JOIN
census AS cen
ON l.state = cen.state_abbr;

```

```

--Create primary key for referential integrity within the database
ALTER TABLE telecom_master
    ADD CONSTRAINT telecom_master_pk
        PRIMARY KEY (customer_id);

```

--DATA EXPLORATION & FEATURE PREPARATION

```

SELECT state_name, population2019, AVG(monthly_charge) AS average_charge,
COUNT(customer_id) AS num_customers
FROM telecom_master
WHERE churn = 'Yes'
GROUP BY 1, 2
HAVING AVG(tenure) < 33
ORDER BY 1;

```

```

SELECT state_name, population2019, AVG(monthly_charge), COUNT(customer_id) AS
num_customers
FROM telecom_master

```

```
WHERE churn = 'No'
GROUP BY 1, 2
HAVING AVG(tenure) > 33
ORDER BY 1;
```

--Customers who have churned with location information:

```
SELECT latitude, longitude, children, age, income, marital,
       churn, gender, tenure, monthly_charge, bandwidth_gb_year,
       contract_type, zip, city, state, county, region, division,
       state_name, population2019
FROM telecom_master
WHERE churn = 'Yes';
```

--Customers who have not churned with location information:

```
SELECT latitude, longitude, children, age, income, marital,
       churn, gender, tenure, monthly_charge, bandwidth_gb_year,
       contract_type, zip, city, state, county, region, division,
       state_name, population2019
FROM telecom_master
WHERE churn = 'No';
```

--ADD column region_name to telecom master:

```
ALTER TABLE telecom_master
ADD COLUMN region_name varchar(10);
```

--Fill values based on region values mapping to names:

```
UPDATE telecom_master
SET region_name =
CASE
  WHEN region = 0 THEN 'PuertoRico'
  WHEN region = 1 THEN 'Northeast'
  WHEN region = 2 THEN 'Midwest'
  WHEN region = 3 THEN 'South'
  WHEN region = 4 THEN 'West'
END;
```

--ADD column division_name to telecom master:

```
ALTER TABLE telecom_master
```

```
ADD COLUMN division_name varchar(30);
```

```
UPDATE telecom_master
SET division_name =
CASE
  WHEN division = 0 THEN 'Puerto Rico'
  WHEN division = 1 THEN 'New England'
  WHEN division = 2 THEN 'Middle Atlantic'
  WHEN division = 3 THEN 'East North Central'
  WHEN division = 4 THEN 'West North Central'
  WHEN division = 5 THEN 'South Atlantic'
  WHEN division = 6 THEN 'East South Central'
  WHEN division = 7 THEN 'West South Central'
  WHEN division = 8 THEN 'Mountain'
  WHEN division = 9 THEN 'Pacific'
END;
```

--Create lifetime_value column based on tenure * monthly_charge per customer.

```
ALTER TABLE telecom_master
ADD COLUMN lifetime_value numeric;
```

```
UPDATE telecom_master
SET lifetime_value = (tenure * monthly_charge);
```

--Top customer per region:

```
SELECT region_name, MAX(lifetime_value) AS top_customer,
RANK() OVER(PARTITION BY region_name)
FROM telecom_master
GROUP BY region_name
ORDER BY top_customer DESC;
```

	region_name character varying (10)	top_customer numeric	rank bigint
1	South	20132.155509140	1
2	Midwest	19726.524860292	1
3	Northeast	19234.06329933	1
4	West	18783.90702539	1
5	PuertoRico	14729.314444171	1

--Top customers per state:

```
SELECT state_name, MAX(lifetime_value) AS top_customer,
RANK() OVER(PARTITION BY state_name)
```

```
FROM telecom_master
GROUP BY state_name
ORDER BY top_customer DESC;
```

	state_name character varying (30)	top_customer numeric	rank bigint
1	South Carolina	20132.155509140	1
2	Florida	19886.710203495	1
3	Minnesota	19726.524860292	1
4	Indiana	19544.508369750	1
5	Texas	19506.251389968	1
6	Mississippi	19484.491073544	1
7	Kentucky	19281.285225500	1

```
--SUMMARIZE lifetime value:
--Compute the min(), avg(), max(), and stddev() of lifetime_value
```

```
SELECT ROUND(MIN(lifetime_value),2) as min_lifetime_value,
       ROUND(MAX(lifetime_value),2) as max_lifetime_value,
       ROUND(AVG(lifetime_value),2) as average_lifetime_value,
       ROUND(STDDEV(lifetime_value),2) as std_lifetime_value
FROM telecom_master;
```

```
--Create a data set for the dashboard that summarizes by state with row number for count
--save as state_lfval_summary.csv for use in tableau
```

```
--Compute the min(), avg(), max(), and stddev() of lifetime_value
SELECT state_name, population2019,
       ROUND(SUM(lifetime_value),2) as total_lifetime_value,
       ROUND(MIN(lifetime_value),2) as min_lifetime_value,
       ROUND(MAX(lifetime_value),2) as max_lifetime_value,
       ROUND(AVG(lifetime_value),2) as average_lifetime_value,
       ROUND(STDDEV(lifetime_value),2) as std_lifetime_value,
       ROW_NUMBER() OVER(PARTITION BY state_name)
FROM telecom_master
GROUP BY state_name, population2019
ORDER BY total_lifetime_value DESC;
```

```
--Make it easier to split the data by tenure type HIGH/LOW based on relation to 33 months.
```

```
ALTER TABLE telecom_master
ADD COLUMN tenure_type varchar(10);
```



```
UPDATE telecom_master
SET tenure_type =
CASE
  WHEN tenure < 33 THEN 'Low'
  WHEN tenure >= 33 THEN 'High'
END;
```