

# SQL Projects

## ❖ Project 1: “Create a Table”

### ➤ Code:

```
CREATE TABLE friends (
```

```
  id INTEGER,
```

```
  name TEXT,
```

```
  birthday DATE );
```

```
INSERT INTO friends (id, name, birthday)
```

```
VALUES (1, 'Ororo Munroe', 1940-05-30);
```

```
SELECT *
```

```
FROM friends;
```

```
INSERT INTO friends (id, name, birthday)
```

```
VALUES (2, 'Neera', '1990-10-04');
```

```
INSERT INTO friends (id, name, birthday)
```

```
VALUES (3, 'Liyana', '2019-12-22');
```

```
UPDATE friends
```

```
SET name = 'Storm'
```

```
WHERE id = 1;
```

```
ALTER TABLE friends
```

```
ADD COLUMN email TEXT;
```

```
UPDATE friends
```

```
SET email = 'storm@codecademy.com'
```

```
WHERE id = 1;
```

```
DELETE FROM friends
```

```
WHERE id = 1;
```

```
SELECT *
```

```
FROM friends;
```

## ❖ Project 2: “New York Restaurants”

```
SELECT DISTINCT neighborhood
FROM nomnom;
SELECT DISTINCT cuisine
FROM nomnom;
```

```
SELECT *
FROM nomnom
WHERE cuisine = 'Chinese';
SELECT *
FROM nomnom
WHERE review >= 4;
```

```
SELECT *
FROM nomnom
WHERE cuisine = 'Italian' AND price = '$$$';
```

```
SELECT *
FROM nomnom
WHERE name LIKE '%meatball%';
```

```
SELECT *
FROM nomnom
WHERE neighborhood IN ('Midtown', 'Downtown', 'Chinatown');
```

```
SELECT *
FROM nomnom
WHERE health IS NULL;
```

```
SELECT name
FROM nomnom
ORDER BY review DESC
LIMIT 10;
```

```
SELECT name, review,
CASE
  WHEN review > 4.5 THEN "Extraordinary"
  WHEN review > 4 THEN "Excellent"
  WHEN review > 3 THEN "Good"
```

```
    WHEN review > 2 THEN "Fair"
    ELSE "Poor"
  END "Review"
FROM nomnom;
```

### ❖ Project 3: “Analyze Hacker News Trends”

**Skills Used:** These queries demonstrate various SQL skills, including data retrieval, aggregation, filtering, calculations, and the use of functions like SUM, COUNT, and strftime.

```
SELECT title, score
FROM hacker_news
ORDER BY score DESC
LIMIT 5;
```

```
SELECT SUM(score) FROM hacker_news;
```

```
SELECT user, SUM(score) FROM hacker_news GROUP BY user HAVING SUM(score) >
200 ORDER BY 2 DESC;
```

```
SELECT (517 + 309 + 304 + 282) / 6366.0;
```

```
SELECT user, COUNT(*) FROM hacker_news WHERE url LIKE
'%watch?v=dQw4w9WgXcQ%' GROUP BY user;
```

```
SELECT CASE
  WHEN url LIKE '%github.com%' THEN 'GitHub'
  WHEN url LIKE '%medium.com%' THEN 'Medium'
  WHEN url LIKE '%nytimes.com%' THEN 'New York Times'
  ELSE 'Other'
END AS 'Source', COUNT(*)
FROM hacker_news
GROUP BY 1;
```

```
SELECT timestamp
FROM hacker_news
LIMIT 10;
```

```
SELECT timestamp,
  strftime('%H', timestamp)
FROM hacker_news
```

```
GROUP BY 1  
LIMIT 20;
```

```
SELECT strftime('%H', timestamp) AS 'Hour',  
       ROUND(AVG(score), 1) AS 'Average Score',  
       COUNT(*) AS 'Number of Stories'  
FROM hacker_news  
WHERE timestamp IS NOT NULL  
GROUP BY 1;
```

## ❖ Project 4: "Lyft Trip Data"

```
SELECT * FROM trips;
```

```
SELECT * FROM riders;
```

```
SELECT * FROM cars;
```

```
SELECT riders.first,
```

```
       riders.last,
```

```
       cars.model
```

```
FROM riders, cars;
```

```
SELECT *
```

```
FROM trips
```

```
LEFT JOIN riders
```

```
    ON trips.rider_id = riders.id;
```

```
SELECT *
```

```
FROM trips
```

```
JOIN cars
```

```
    ON trips.car_id = cars.id;
```

```
SELECT *
```

```
FROM riders
```

```
UNION
```

```
SELECT *
```

```
FROM riders2;
```

```
SELECT AVG(cost)
```

```
FROM trips;
```

```
SELECT *
```

```
FROM riders
```

```
WHERE total_trips < 500;
```

```
SELECT COUNT(*)
```

```
FROM cars
```

```
WHERE status = 'active';
```

```
SELECT *
```

```
FROM cars
```

```
ORDER BY trips_completed DESC
```

```
LIMIT 2;
```



## ❖ Project 5: “Covid 19 Data Exploration”

```
/*  
Covid 19 Data Exploration
```

Skills used: Joins, CTE's, Temp Tables, Windows Functions, Aggregate Functions, Creating Views, Converting Data Types

```
*/
```

```
Select *  
From PortfolioProject.CovidDeaths  
Where continent is not null  
order by 3,4
```

-- Select Data that we are going to be starting with

```
SELECT Location, date, total_cases, new_cases, total_deaths,  
population  
From PortfolioProject..CovidDeaths  
Where continent is not null  
order by 1,2
```

-- Total Cases vs Total Deaths

-- Shows likelihood of dying if you contract covid in your country

```
Select Location, date, total_cases,total_deaths,  
(total_deaths/total_cases)*100 as DeathPercentage  
From PortfolioProject..CovidDeaths  
Where location like '%states%'  
and continent is not null  
order by 1,2
```

-- Total Cases vs Population

-- Shows what percentage of population infected with Covid

```
Select Location, date, Population, total_cases,  
(total_cases/population)*100 as PercentPopulationInfected  
From PortfolioProject..CovidDeaths  
--Where location like '%states%'  
order by 1,2
```

-- Countries with Highest Infection Rate compared to Population

```
Select Location, Population, MAX(total_cases) as  
HighestInfectionCount, Max((total_cases/population))*100 as  
PercentPopulationInfected  
From PortfolioProject..CovidDeaths  
--Where location like '%states%'  
Group by Location, Population  
order by PercentPopulationInfected desc
```

-- Countries with Highest Death Count per Population

```
Select Location, MAX(cast(Total_deaths as int)) as TotalDeathCount  
From PortfolioProject..CovidDeaths  
--Where location like '%states%'  
Where continent is not null  
Group by Location  
order by TotalDeathCount desc
```

-- BREAKING THINGS DOWN BY CONTINENT

-- Showing continents with the highest death count per population

```
Select continent, MAX(cast(Total_deaths as int)) as TotalDeathCount  
From PortfolioProject..CovidDeaths  
--Where location like '%states%'  
Where continent is not null  
Group by continent  
order by TotalDeathCount desc
```

-- GLOBAL NUMBERS

```
Select SUM(new_cases) as total_cases, SUM(cast(new_deaths as
int)) as total_deaths, SUM(cast(new_deaths as
int))/SUM(New_Cases)*100 as DeathPercentage
From PortfolioProject..CovidDeaths
--Where location like '%states%'
where continent is not null
--Group By date
order by 1,2
```

-- Total Population vs Vaccinations  
-- Shows Percentage of Population that has recieved at least one  
Covid Vaccine

```
Select dea.continent, dea.location, dea.date, dea.population,
vac.new_vaccinations
, SUM(CONVERT(int,vac.new_vaccinations)) OVER (Partition by
dea.Location Order by dea.location, dea.Date) as
RollingPeopleVaccinated
--, (RollingPeopleVaccinated/population)*100
From PortfolioProject..CovidDeaths dea
Join PortfolioProject..CovidVaccinations vac
On dea.location = vac.location
and dea.date = vac.date
where dea.continent is not null
order by 2,3
```

-- Using CTE to perform Calculation on Partition By in previous query

```
With PopvsVac (Continent, Location, Date, Population,
New_Vaccinations, RollingPeopleVaccinated)
as
(
Select dea.continent, dea.location, dea.date, dea.population,
vac.new_vaccinations
```

```

, SUM(CONVERT(int,vac.new_vaccinations)) OVER (Partition by
dea.Location Order by dea.location, dea.Date) as
RollingPeopleVaccinated
--, (RollingPeopleVaccinated/population)*100
From PortfolioProject..CovidDeaths dea
Join PortfolioProject..CovidVaccinations vac
    On dea.location = vac.location
    and dea.date = vac.date
where dea.continent is not null
--order by 2,3
)
Select *, (RollingPeopleVaccinated/Population)*100
From PopvsVac

```

-- Using Temp Table to perform Calculation on Partition By in previous query

```

DROP Table if exists #PercentPopulationVaccinated
Create Table #PercentPopulationVaccinated
(
Continent nvarchar(255),
Location nvarchar(255),
Date datetime,
Population numeric,
New_vaccinations numeric,
RollingPeopleVaccinated numeric
)

```

```

Insert into #PercentPopulationVaccinated
Select dea.continent, dea.location, dea.date, dea.population,
vac.new_vaccinations
, SUM(CONVERT(int,vac.new_vaccinations)) OVER (Partition by
dea.Location Order by dea.location, dea.Date) as
RollingPeopleVaccinated
--, (RollingPeopleVaccinated/population)*100
From PortfolioProject..CovidDeaths dea
Join PortfolioProject..CovidVaccinations vac
    On dea.location = vac.location
    and dea.date = vac.date

```

```
--where dea.continent is not null
--order by 2,3
```

```
Select *, (RollingPeopleVaccinated/Population)*100
From #PercentPopulationVaccinated
```

```
-- Creating View to store data for later visualizations
```

```
Create View PercentPopulationVaccinated as
Select dea.continent, dea.location, dea.date, dea.population,
vac.new_vaccinations
, SUM(CONVERT(int,vac.new_vaccinations)) OVER (Partition by
dea.Location Order by dea.location, dea.Date) as
RollingPeopleVaccinated
--, (RollingPeopleVaccinated/population)*100
From PortfolioProject..CovidDeaths dea
Join PortfolioProject..CovidVaccinations vac
    On dea.location = vac.location
    and dea.date = vac.date
where dea.continent is not null
```