Lab-5B

Constraints allow the designer to control column values. Primary and foreign keys are types of constraints. The CHECK constraint insures that the column values falls with a specified range. The UNIQUE constraint insures that no duplicates are created. The NOT NULL constraint insures that a value is entered in every row. Constraints can be entered as column constraints or table constraints. Column constraints only apply to the column and appear after the column name. Table Constraints appear at the end of the table definition.

The PRIMARY KEY constraint insures that there is a value entered in every row and no two are alike (UNIQUE). Primary keys enable tables to relate to each other with *referential integrity*. You don't have to create a separate column for a primary key. Any column that meets the above criteria (unique and not null) can server as a , so called 'natural', primary key. When you use a new column for your key, it is called a 'surrogate' key. I prefer the surrogate key because you can use the system to create and keep track of the keys.

Following are two ways to create a natural key. NOTE: if you were to create a primary key involving two columns (it is called a composite key) you'd have to use the second method - table constraint.

```
CREATE TABLE natural_key_example(
  license_id varchar(10) CONSTRAINT license_key PRIMARY KEY,
  first_name varchar(50),
  last_name varchar(50)
);
CREATE TABLE natural_key_example(
  license_id varchar(10) ,
  first_name varchar(50),
  last_name varchar(50)
  CONSTRAINT license_key PRIMARY KEY (license_id)
);
```

Create either table and then execute the following two INSERT statements and see what happens.

```
INSERT INTO natural_key_example(license_id, first_name, last_name)
VALUES ('PA46522', 'Bill', 'Maloney') # note the single quotes around text)
INSERT INTO natural_key_example(license_id, first_name, last_name)
VALUES ('PA46522', 'Jane', 'Jefferson') # note the single quotes around text)
```

Here we will create an autoincrementing surrogate key.

```
CREATE TABLE surrogate_key_example (
  order_number bigserial,
  product_name varchar(50),
```

```
  order_date date,
  CONSTRAINT order_key PRIMARY KEY (order_number)
);
```

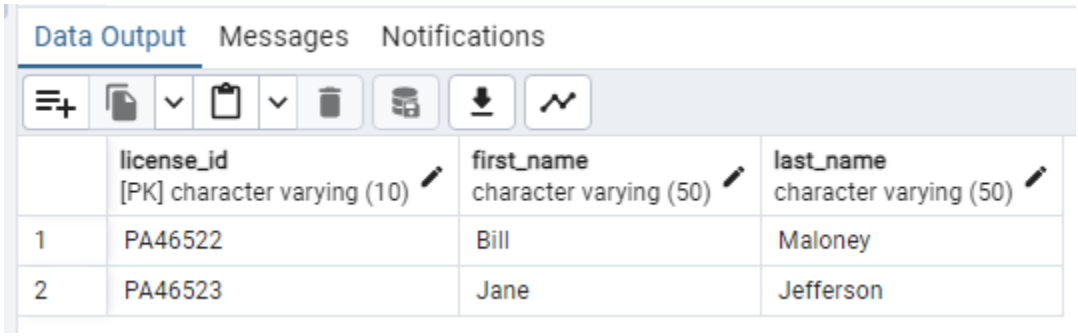
And, then insert some data.

```
INSERT INTO surrogate_key_example(product_name, order_date)
VALUES ('Shoe Scuffer', '12/3/2022') # note the single quotes around date),
       ('Tire Unbalancer', '1/3/2023')
       ('Stain Enhancer', '8/7/2021');
```

Use a SELECT query to see this result.

Create two tables. One should contain a natural key and the other surrogate key. Add some data and query the result (it never hurts to experiment with your queries. Date queries are perfect for BETWEEN conditions, for example).

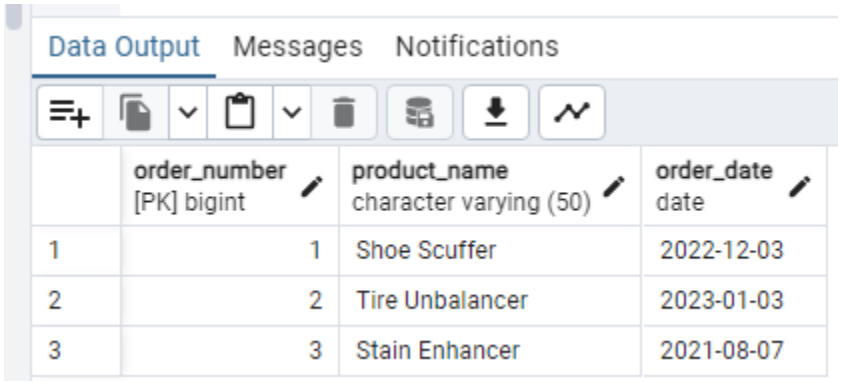
Natural_key table:



Data Output | Messages | Notifications

	license_id [PK] character varying (10)	first_name character varying (50)	last_name character varying (50)
1	PA46522	Bill	Maloney
2	PA46523	Jane	Jefferson

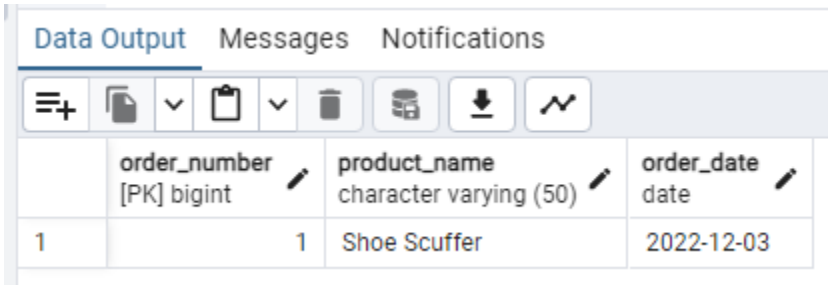
Surrogate_key table:



Data Output | Messages | Notifications

	order_number [PK] bigint	product_name character varying (50)	order_date date
1	1	Shoe Scuffer	2022-12-03
2	2	Tire Unbalancer	2023-01-03
3	3	Stain Enhancer	2021-08-07

Surrogate_key table with BETWEEN condition:



	order_number [PK] bigint	product_name character varying (50)	order_date date
1	1	Shoe Scuffer	2022-12-03