Connecting throguh terminal:

* Switch to local host

“psql –h hostname –p port –U server name”

SELECT STEATEMENT:

SELECT \* FROM person WHERE place\_of\_birth IN (‘China’, ‘Mongol’);

Instead of duplicating the code I have the option of simply writing an IN

And then include all the values I’d want

SELECT \* FROM person WHERE email LIKE ‘%something%’;

Or

The mod sign here means that any character or characters before ‘something’ or after it are ignored and if something was found in an email the condition would be true thus a result will be returned

SELECT \* FROM person WHERE email LIKE ‘\_\_\_\_\_\_\_something’;

Each Dash represents a character here

So the example above lets say we have 5 dashs

So the condition here means bring me anyone with an email that begins with 5 characters followed by ‘something’

SELECT \* FROM person WHERE country\_of\_birth ILIKE ‘m%’

The ILIKE keyword ignores case sensitiveness

Distinct

SELECT DISTINCT country\_of\_birth FROM PERSON;

Gets all the distinct countries in the person table

SELECT country\_of\_birth, COUNT(\*) FROM person GROUP BY country\_of\_birth;

Gets every single country and its users

SQL KEYWORDS

DISTINCT

LIMIT (SPECIFIES HOW MANY ROWS YOU WANT BACK WITH YOUR QUERY)

AS (ALIAS)

ILIKE ( COMPARING RESULT N BRINING IT BACK / NON CASE SENSETIVE)

LIKE ( CASE SENSETIVE)

GROUP BY (GROUPS DATA BY A COLUMN)

ORDER BY ASC DESC ( GROUPS DATA IN AN ASCENDING ORDER OR VICE-VERSA)

SQL AGGREGATE FUNCTIONS

ROUND(NUMBER,DECIMAL POINT)

SUM

MIN

MAX

AVERAGE

COUNT

COALESCE(VALUE (IF NUL), GET THIS VALUE INSTEAD)

NULLIF(10, 10)

NOW() GIVES US TIME STAMP

CASTING IN SQL:

DOUBLE COLON ::

NOW()::DATE;j

NOW()::TIME;

INTERVALS:

SELECT NOW() – INTERVAL ’10 YEARS’ – OR MONTHS OR DAYS OR EACH AND EVERYONE OF THE PREVIOUS BUT WITHOUT THE PLURAL S

EXTRACT()

SELECT EXTRACT(CENTURY MILLISECONDS YEAR MONTH DOW OR DAY AND MANY MORE, NOW());

//DOW = DAY OF THE WEEK

SELECT (NOW() + INTERVAL ’10 MONTHS’)::DATE;

CONSTRAINTS:

* ADDING A CONSTRAINT UPON ON A COLUMN IN THE TABLE CHECK(CONDITION)
* UNIQUE()

EXAMPLE:

ALTER TABLE person ADD CONSTRAINT name\_of\_constraint UNIQUE(column\_name);

ALTER TABLE person ADD CONSTRAINT name\_of\_constraint UNIQUE(column\_name);

INSERT INTO person (id, first\_name, last\_name, gender, email, date\_of\_birth, country\_of\_birth)

VALUES (17 , 'Anabel', 'Farrance', 'Female', 'afarranceg@shop-pro.jp', '2022-06-04', 'China')

ON CONFLICT (id) DO NOTHING;

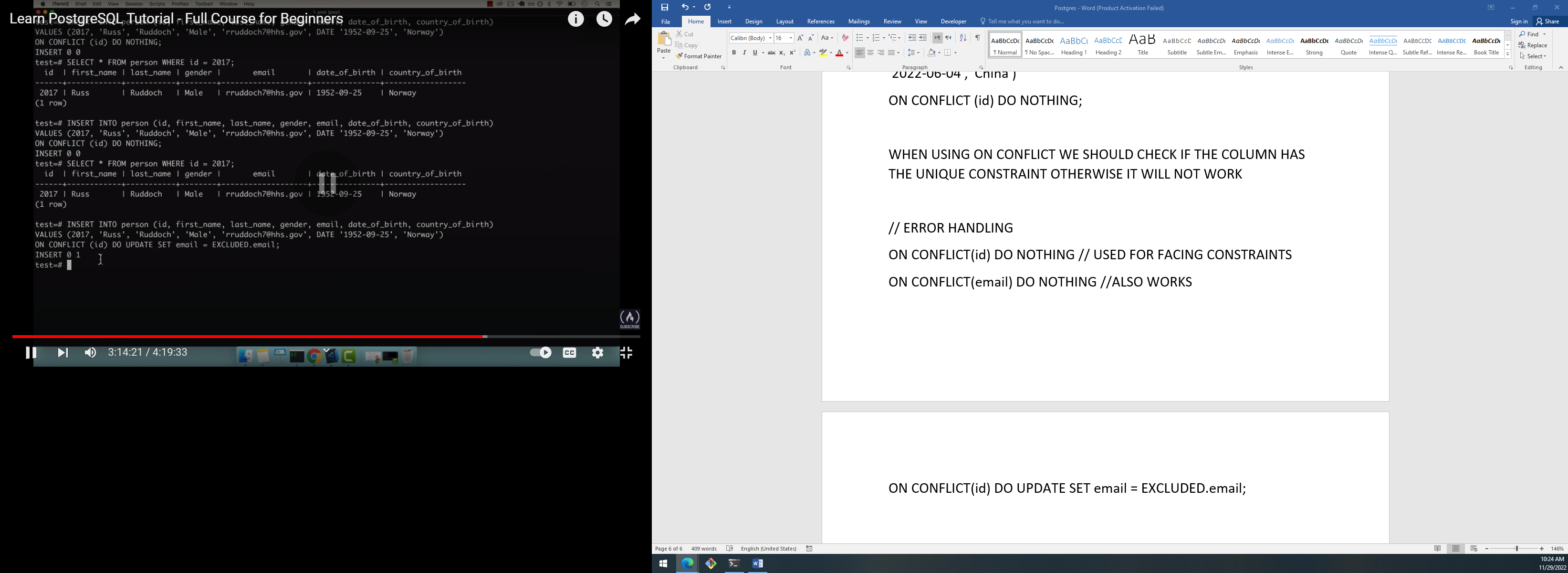
WHEN USING ON CONFLICT WE SHOULD CHECK IF THE COLUMN HAS THE UNIQUE CONSTRAINT OTHERWISE IT WILL NOT WORK

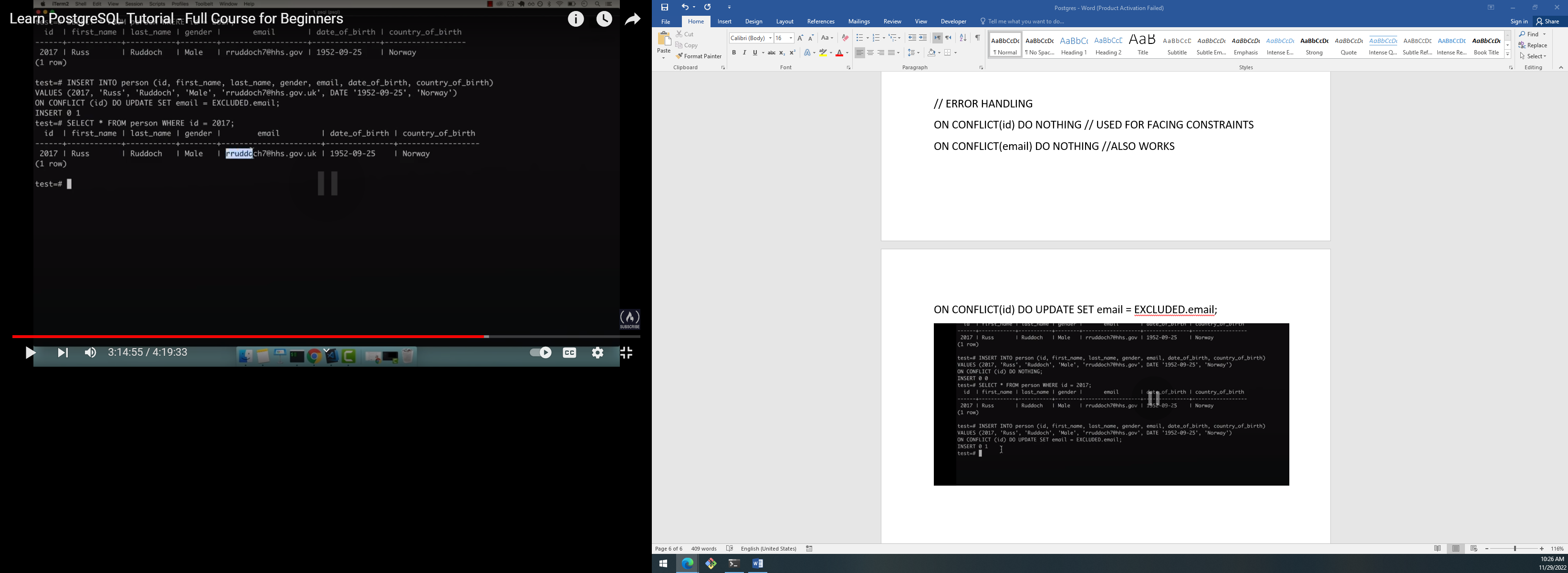
// ERROR HANDLING

ON CONFLICT(id) DO NOTHING // USED FOR FACING CONSTRAINTS

ON CONFLICT(email) DO NOTHING //ALSO WORKS

ON CONFLICT(id) DO UPDATE SET email = EXCLUDED.email;





IT ALSO WORKS ON MULTIPLE COLUMNS

INSERT INTO person (id, first\_name, last\_name, email, gender, date\_of\_birth, country\_of\_birth)

VALUES

(1, 'Ricky', 'Manus', 'rmanus0@purevolume.com.us', 'Female' , DATE '2022-10-02', 'Russia')

ON CONFLICT(id) DO UPDATE SET email = EXCLUDED.email, date\_of\_birth = EXCLUDED.date\_of\_birth;

INSERT 0 1

JOIN 🡪 SELECT \* FROM person

JOIN car ON person.car\_id = car.id;

LEFT JOIN 🡪

SELECT \* FROM PERSON

LEFT JOIN car ON car.id = person.car\_id;

SELECT \* FROM PERSON

LEFT JOIN car ON car.id = person.car\_id

WHERE car.\* IS NULL;

Generate sql files

\copy (SELECT \* FROM PERSON LEFT JOIN car on car.id = person.car\_id) TO 'C:\Users\Public\results.csv' DELIMITER ',' CSV HEADER;

RESTARTING SEQUENCE WITH OR WITHOUT A VALUE

WITH A VALUE 🡪

ALTER SEQUENCE person\_id\_seq RESTART WITH 10;

WITHOUT A VALUE 🡪

ALTER SEQUENCE person\_id\_seq RESTART;