Week 4

Repo <https://github.com/MGIS-UMN/GIS5577_week4>

-- What data type could we use for creating a primary

Serial

-- What statement would help add a serial column.

Alter table

ALTER TABLE candy\_grab

ADD transaction\_id serial;

UPDATE candy\_grab

SET customer\_id = 1556

WHERE candy\_owner = 'David';

-- how many records will be updated

At least 6

16

16

16

SELECT COUNT(1) FROM candy\_grab

WHERE candy\_owner='David';

-- Update the timestamp

UPDATE candy\_grab

SET purchased\_date = '10-31-2012'

WHERE transaction\_id = 2

SELECT \*

FROM candy\_grab

WHERE transaction\_id = 2

Candy\_type

Class

CREATE TABLE table\_one(id integer, value varchar(5));

INSERT INTO table\_one (id, value) VALUES (1, 'a'),(4, 'd'),(2, 'b');

CREATE TABLE table\_two(id integer, value varchar(5));

INSERT INTO table\_two (id, value) VALUES (2, 'x'),(3, 'y'),(5, 'v');

Breakout Group 1:

CREATE TABLE languages(

language\_name character(15),

id\_number serial,

choice int

);

INSERT INTO languages(language\_name, choice)

VALUES ('English', 1), ('Spanish', 2), ('French', 3)

SELECT \*

FROM languages

CREATE TABLE languages\_join(

language\_name character(15),

id\_number serial,

join\_key int

);

INSERT INTO languages\_join(language\_name, join\_key)

VALUES ('English', 1), ('Spanish', 2), ('French', 3), ('English', 4), ('Spanish', 4), ('English', 5),('French', 5), ('Spanish', 6),('French', 6),('English', 7),('French', 7), ('Spanish', 7)

SELECT \*

FROM languages\_join

Cartesian product

select \*

from table\_one, table\_two

--table one

-- 1 "a"

-- 4 "d"

-- 2 "b"

--table two

-- 2 "x"

-- 3 "y"

-- 5 "v"

SELECT \*

FROM table\_one as t1

INNER JOIN table\_two as t2 ON (t1.id = t2.id)

SELECT t1.id, t2.value\*2

FROM table\_one

INNER JOIN table\_two ON (table\_one.id = table\_two.id)

-- LEFT JOIN

SELECT \*

FROM table\_one t1

LEFT JOIN table\_two t2 ON (t1.id = t2.id)

1 "a"

2 "b" 2 "x"

4 "d"

SELECT \*

FROM table\_one t1

RIGHT JOIN table\_two t2 ON (t1.id = t2.id)

SELECT t1.id, unique\_field\_name

FROM table\_one t1 RIGHT JOIN table\_two t2 ON (t2.id = t1.id) -- multiple fields in a join statement and (t2.=)

-- the order of the alias does not matter

t1.id -- [Table].[field] -- any fields that have the same name.