Topics

1. Review
2. Set Operators
3. Using set operators to combine multiple queries into a single query
4. Controlling the order of rows returned
5. Set Operator Rules

Midterm

- Review for Midterm Test is next week
- Midterm Test is the following week in the lab period

Subquery

- What SQL SELECT clauses support subqueries?
- SELECT
  - WHERE
  - HAVING
  - FROM

Subquery in SQL statements

- What other statements support a subquery
- CREATE
- INSERT
- DELETE

SUBQUERY TYPES

- What are the 3 general types of subqueries?
- Single Row Subqueries – returns one row in the “inner” result set
- Multi Row Subqueries – returns multiple rows in the “inner” result set
- Multi-column Subqueries – returns multiple columns in the “inner” result set

SUBQUERY REVIEW

- What job categories in the STAFF table make a higher average salary than DESIGNERS do in the SENECACORP table?
- Also show the high, low and total STAFF salaries for the job types that have a higher average salary then SENECACORP’S DESIGNERS

- Two different queries

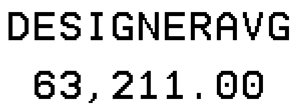
```
SELECT  AVG(SALARY)
FROM    SENECACORP
WHERE   JOB = 'DESIGNER'
```

```
SELECT JOB,
       CAST(AVG(SALARY) AS DECIMAL(7,2)) AVGSALARY,
       MAX(SALARY) HIGHSALARY,
       MIN(SALARY) LOWSALARY,
       SUM(SALARY) TOTALSALARY
FROM   STAFF
GROUP  BY JOB
```

- Which one is the subquery
- Does it go in a Where, From or Having Clause

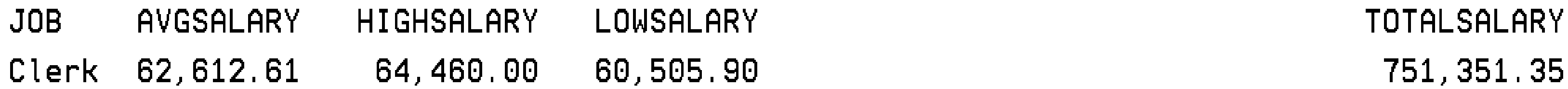
SUBQUERY REVIEW

- Both tables have SALARY and JOB columns.
- You can’t refer to the number shown here



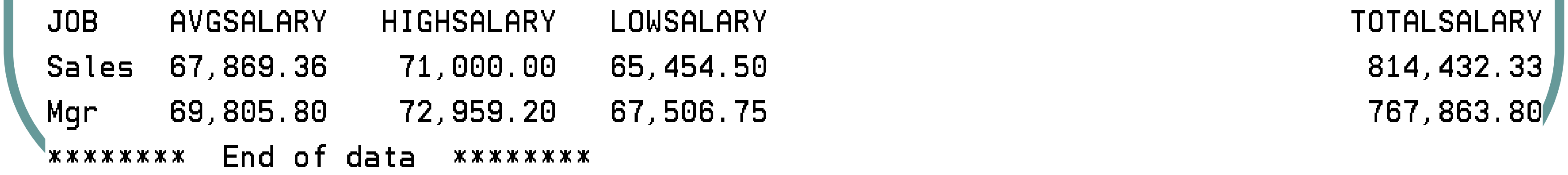
```
DESIGNERAVG
 63,211.00
```

- Your SQL statement needs to figure this out
- It looks like this low average will not be included



```
JOB      AVGSALARY    HIGHSALARY    LOWSALARY                                  TOTALSALARY
Clerk    62,612.61     64,460.00    60,505.90                                   751,351.35
```

- These averages will be included

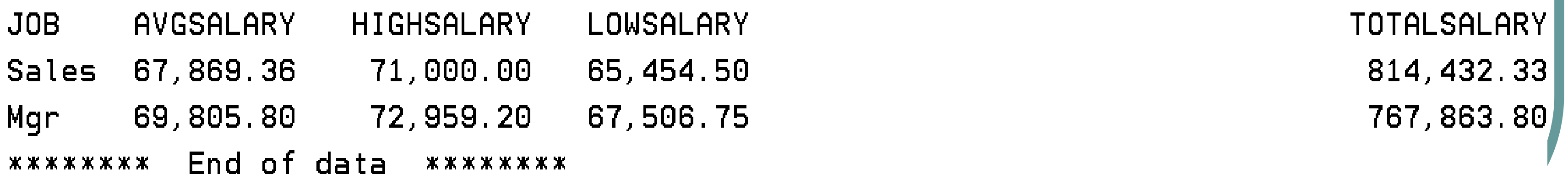


```
JOB      AVGSALARY    HIGHSALARY    LOWSALARY                                  TOTALSALARY
Sales    67,869.36     71,000.00    65,454.50                                   814,432.33
Mgr      69,805.80     72,959.20    67,506.75                                   767,863.80
********   End of data   ********
```

SUBQUERY REVIEW

Does the subquery go in a Where, From or Having Clause

```
SELECT JOB,
       CAST(AVG(SALARY) AS DECIMAL(7,2)) AVGSALARY,
       MAX(SALARY) HIGHSALARY,
       MIN(SALARY) LOWSALARY,
       SUM(SALARY) TOTALSALARY
FROM   STAFF
GROUP BY JOB
HAVING AVG(SALARY) >
       (SELECT AVG(SALARY) FROM SENECACORP
        WHERE JOB = 'DESIGNER')
```



```
JOB     AVGSALARY    HIGHSALARY    LOWSALARY                              TOTALSALARY
Sales   67,869.36     71,000.00    65,454.50                               814,432.33
Mgr     69,805.80     72,959.20    67,506.75                               767,863.80
********  End of data  ********
```

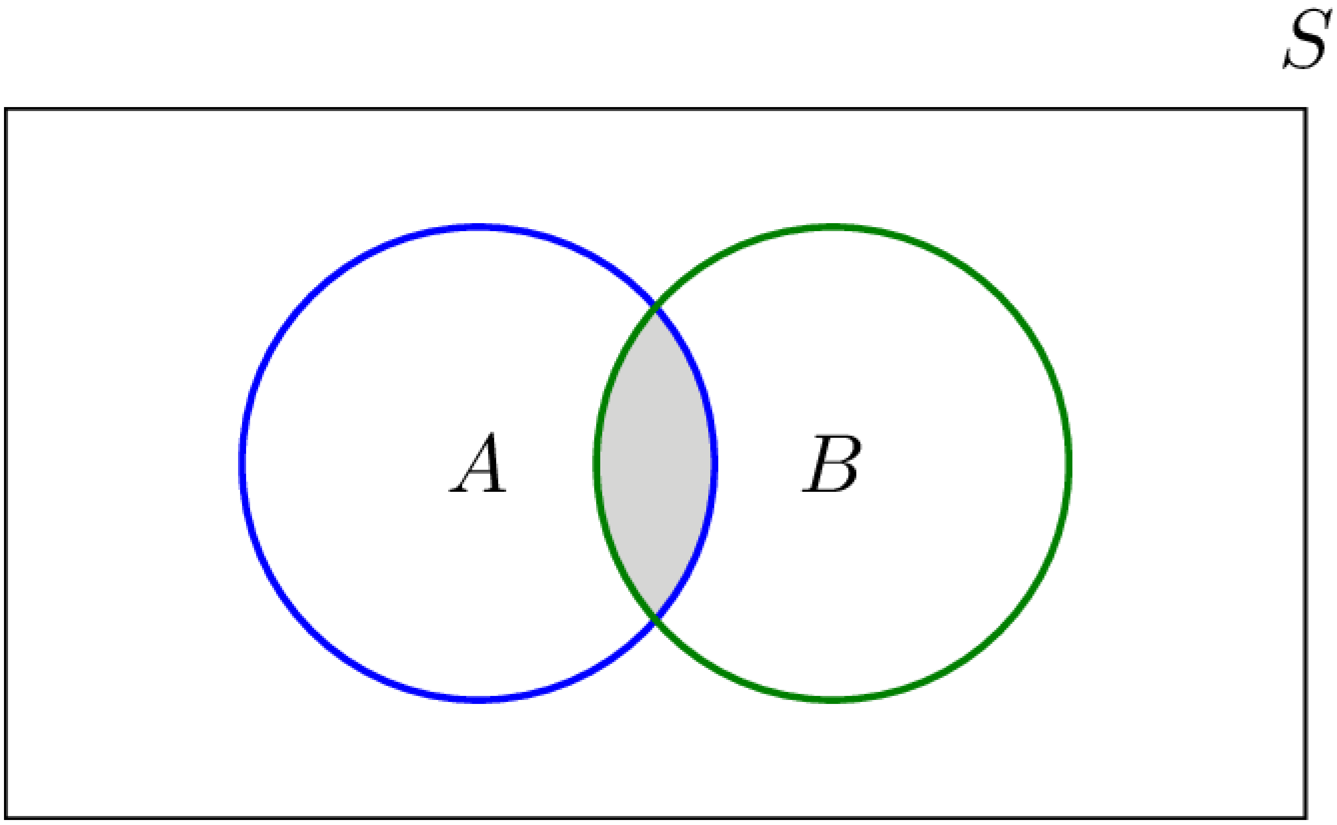
Set Theory – a Mathematical construct

- A {2,4,6,8,10}
- B {1,2,3,4}
- A ∩ B A intersect B
- A ∩ B {2,4}
- A U B Union of A and B
- A U B {1,2,3,4,6,8,10}
- Or {1,2,2,3,4,4,6,8,10}

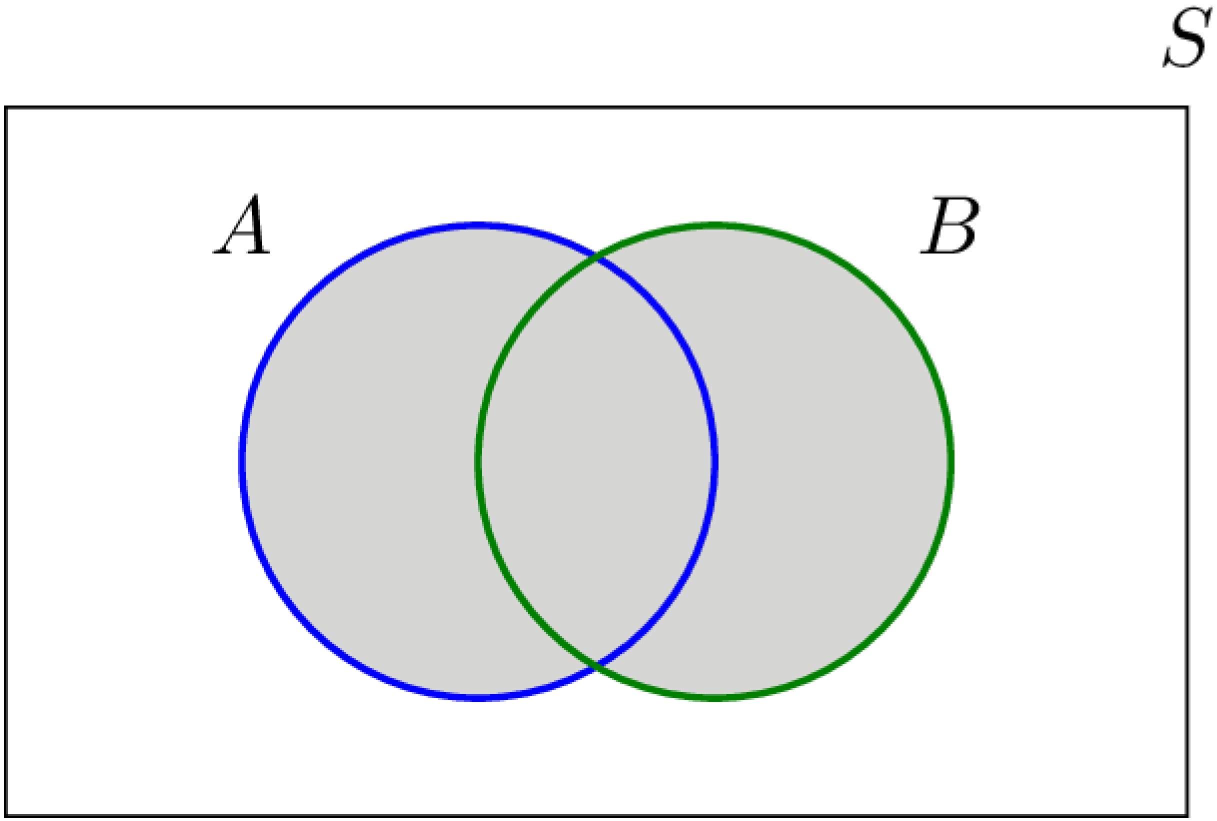
Set Theory – a Mathematical construct

- A {2,4,6,8,10}
- B{1,2,3,4}
- A - B A minus B
- A - B {6,8,10}

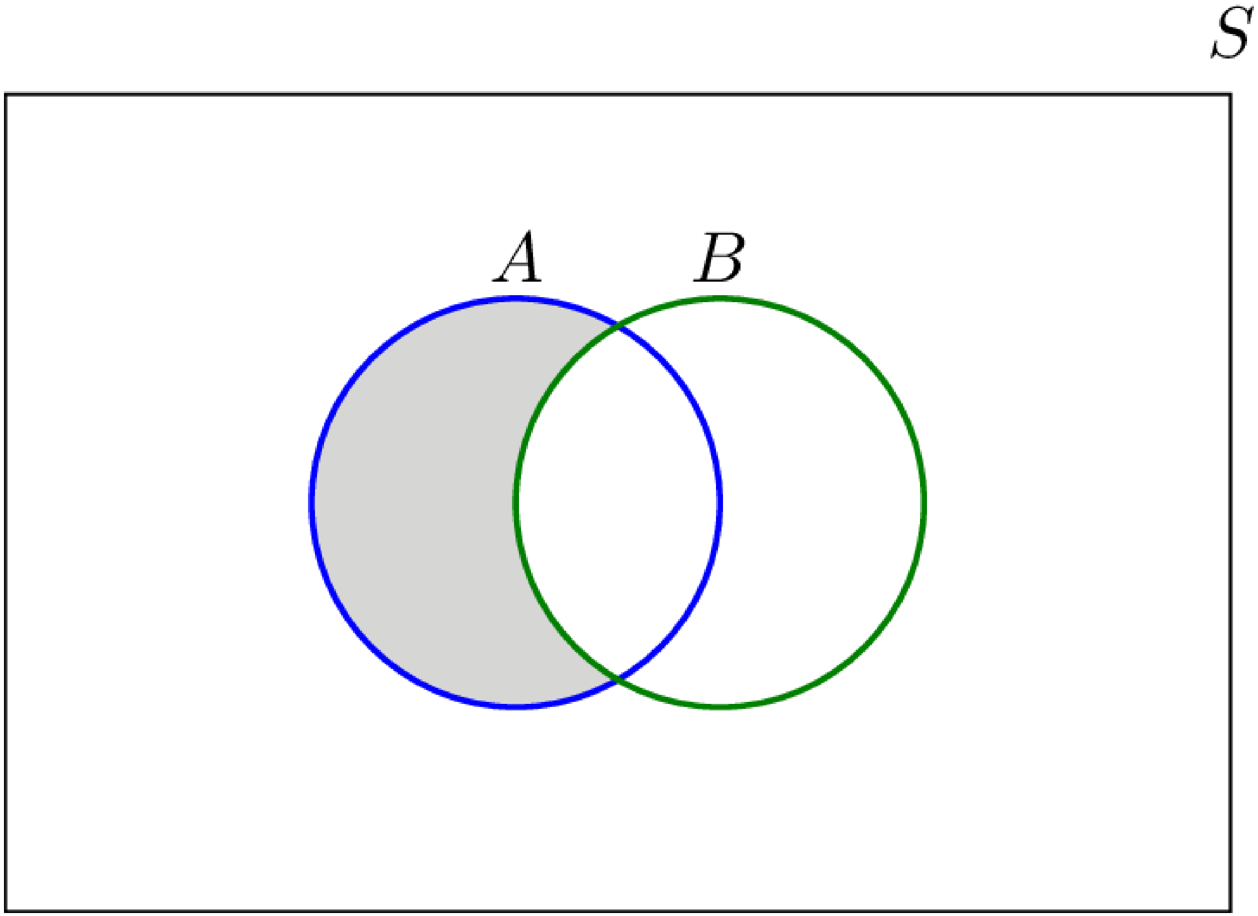
Venn Diagram for Intersection

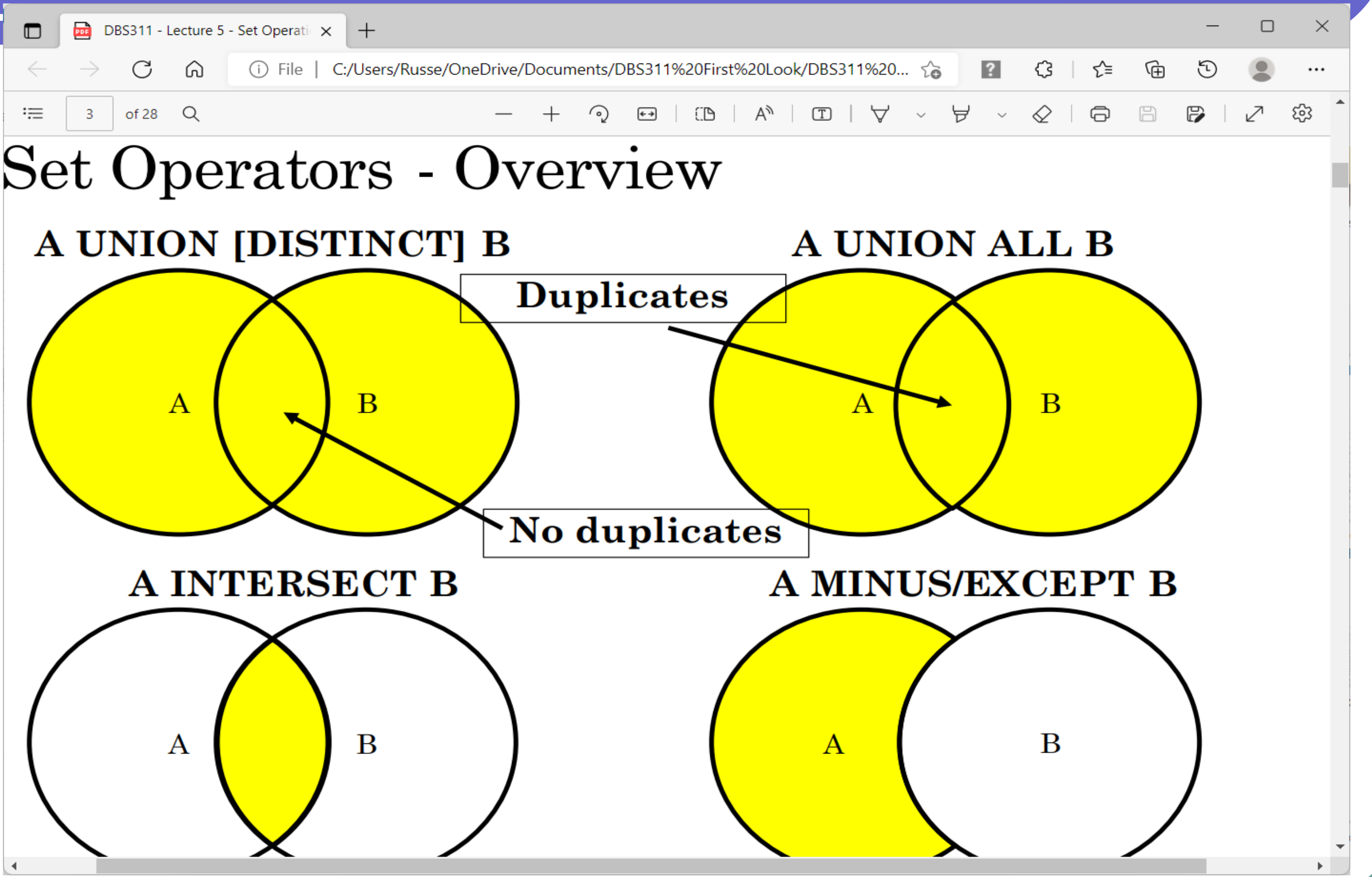


Venn Diagram for Union



Venn Diagram for Minus / Except




DBS311 - Lecture 5 - Set Operati
File | C:/Users/Russe/OneDrive/Documents/DBS311%20First%20Look/DBS311%20...
3
of 28

Set Operators - Overview

A UNION [DISTINCT] B

A UNION ALL B

Duplicates

No duplicates

A INTERSECT B

A MINUS/EXCEPT B

Set Operators

- UNION – returns all rows across both tables, excluding duplicates
- UNION ALL – returns all rows across both tables, including duplicates – can get a different sort order than UNION gives
- INTERSECT – returns common rows across both tables, excluding duplicates
- INTERSECT ALL – returns common rows across both tables, including duplicates
- DB2 EXCEPT – returns rows unique to “LEFT” table, excluding duplicates
- DB2 EXCEPT ALL – returns rows unique to “LEFT” table, including duplicates
- Oracle uses MINUS instead of EXCEPT

Set Operators – Data Sets

- DEPT

```
DEPTNO  DEPTNAME         MGRNO  ADMRDEPT  LOCATION
  500   SALESEAST           3      502    TORONTO
  501   SALESWEST           5      502    VANCOUVER
  502   ADMIN               8      502    TORONTO
  503   HR                  8      502    TORONTO
********  End of data  ********
```

- DEPTUS

```
DEPTNO  DEPTNAME         MGRNO  ADMRDEPT  LOCATION
  600   SALESEASTUS        11      502    NEW YORK
  601   SALESWESTUS        19      502    LOS ANGELES
  502   ADMIN               8      502    TORONTO
  503   HR                  8      502    TORONTO
********  End of data  ********
```

Set Operators – UNION

- UNION – returns all rows across both tables, excluding duplicates

```
SELECT DEPTNO, DEPTNAME
FROM   DEPT
UNION
SELECT DEPTNO, DEPTNAME
FROM   DEPTUS
```

```
DEPTNO   DEPTNAME
  502    ADMIN
  501    SALESWEST
  601    SALESWESTUS
  500    SALESEAST
  600    SALESEASTUS
  503    HR
********   End of data
```

Set Operators – UNION

- UNION ALL – returns all rows across both tables, including duplicates

```
SELECT DEPTNO, DEPTNAME
FROM   DEPT
UNION ALL
SELECT DEPTNO, DEPTNAME
FROM   DEPTUS
```

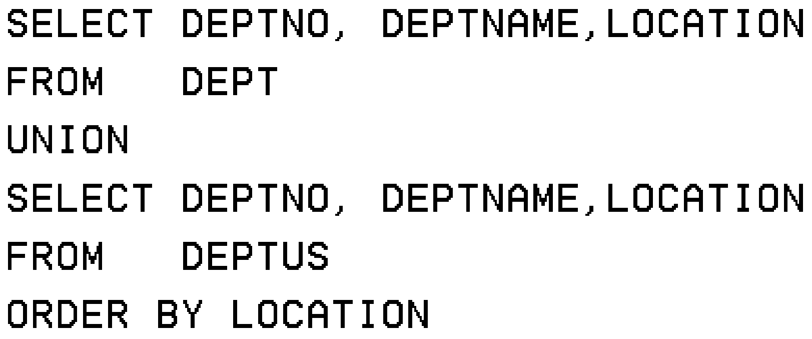
```
DEPTNO   DEPTNAME
  500    SALESEAST
  501    SALESWEST
  502    ADMIN
  503    HR
  600    SALESEASTUS
  601    SALESWESTUS
  502    ADMIN
  503    HR
```

Set Operators – UNION

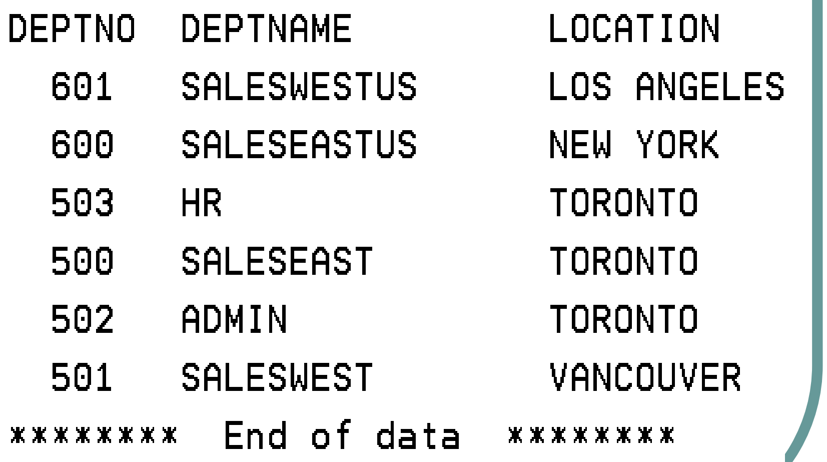
- Our versions of DB2 and Oracle support ALL for UNION
- Our versions of DB2 and Oracle do not support ALL for INTERSECT, MINUS (EXCEPT)
- Later versions of DB2 and Oracle do support ALL for the above, we just do not have the latest versions installed.

Set Operators – UNION – with ORDER BY

When you use UNION Any ORDER BY clause must appear after the last subselect that is part of the union.



```
SELECT DEPTNO, DEPTNAME,LOCATION
FROM   DEPT
UNION
SELECT DEPTNO, DEPTNAME,LOCATION
FROM   DEPTUS
ORDER BY LOCATION
```



```
DEPTNO  DEPTNAME         LOCATION
  601   SALESWESTUS      LOS ANGELES
  600   SALESEASTUS      NEW YORK
  503   HR               TORONTO
  500   SALESEAST        TORONTO
  502   ADMIN            TORONTO
  501   SALESWEST        VANCOUVER
********  End of data  ********
```

Set Operators – INTERSECT

- INTERSECT – returns common rows across both tables, excluding duplicates

```
SELECT DEPTNO,DEPTNAME,LOCATION
FROM DEPT
INTERSECT
SELECT DEPTNO,DEPTNAME,LOCATION
FROM DEPTUS
```

```
DEPTNO  DEPTNAME                 LOCATION
  503   HR                       TORONTO
  502   ADMIN                    TORONTO
********  End of data  ********
```

Set Operators – INTERSECT

- What happens if we change TORONTO to Toronto in the DEPTUS DeptNo 503 row?

```
 UPDATE DEPTUS
 SET LOCATION = 'Toronto'
 WHERE DEPTNO = 503
```

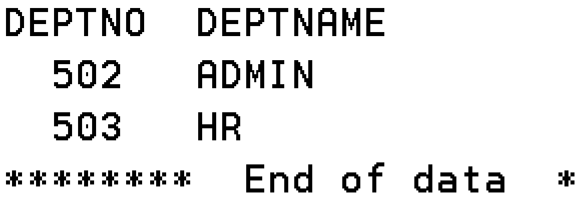
```
SELECT DEPTNO, DEPTNAME,LOCATION
FROM   DEPT
INTERSECT
SELECT DEPTNO, DEPTNAME,LOCATION
FROM   DEPTUS
```

```
                    DEPTNO  DEPTNAME          LOCATION
                      502   ADMIN             TORONTO
                    ********  End of data  ********
```

Set Operators – INTERSECT

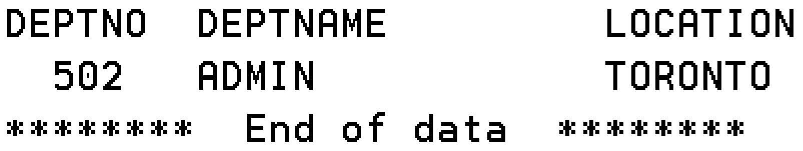
- What happens if we don't include LOCATION in the previous statement?

```
SELECT DEPTNO,DEPTNAME
FROM DEPT
INTERSECT
SELECT DEPTNO,DEPTNAME
FROM DEPTUS
```



```
DEPTNO   DEPTNAME
  502    ADMIN
  503    HR
********   End of data   *
```

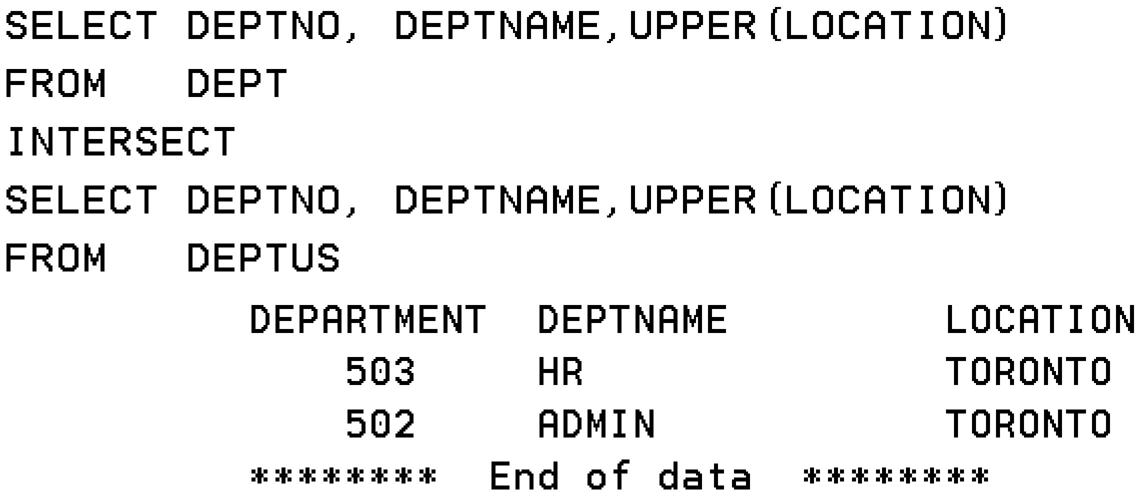
- Instead of



```
DEPTNO   DEPTNAME                LOCATION
  502    ADMIN                   TORONTO
********   End of data   ********
```

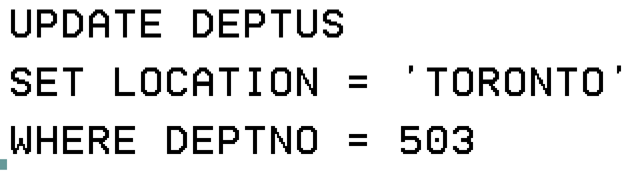
Set Operators – INTERSECT

- Before resetting back to TORONTO, is there any other way to get both rows to INTERSECT again and show location?



```
SELECT DEPTNO, DEPTNAME,UPPER(LOCATION)
FROM   DEPT
INTERSECT
SELECT DEPTNO, DEPTNAME,UPPER(LOCATION)
FROM   DEPTUS
```

```
        DEPARTMENT  DEPTNAME          LOCATION
            503     HR                TORONTO
            502     ADMIN             TORONTO
        ********  End of data  ********
```



```
UPDATE DEPTUS
SET LOCATION = 'TORONTO'
WHERE DEPTNO = 503
```

Set Operators – INTERSECT

```
SELECT DEPTNO,DEPTNAME,LOCATION
FROM DEPT
INTERSECT
SELECT DEPTNO,DEPTNAME
FROM DEPTUS
```

- This produces an error

```
Number of columns not consistent.
```

- For Set Operations, the columns after SELECT must be:
  - The same columns in each SELECT
  - With compatible data types
    - If data types are different, you may need to CAST

Set Operators – INTERSECT – with AS

```
SELECT DEPTNO, DEPTNAME,LOCATION AS X
FROM   DEPT
INTERSECT
SELECT DEPTNO, DEPTNAME,LOCATION AS Y
FROM   DEPTUS
```

- Is the column heading X or Y for LOCATION?

```
DEPTNO  DEPTNAME          X
  503   HR                TORONTO
  502   ADMIN             TORONTO
```

Set Operators – EXCEPT / MINUS

- EXCEPT – returns rows unique to the "LEFT" or first table, excluding duplicates

```
SELECT DEPTNO,
       DEPTNAME,
       LOCATION
FROM   DEPT
EXCEPT
SELECT DEPTNO,
       DEPTNAME,
       LOCATION
FROM   DEPTUS
ORDER BY DEPTNO
```

```
DEPTNO   DEPTNAME          LOCATION
  500    SALESEAST         TORONTO
  501    SALESWEST         VANCOUVER
```

Set Operators – EXCEPT / MINUS

- EXCEPT – What happens if you change the order of the tables?

```
  SELECT DEPTNO,
         DEPTNAME,
         LOCATION
  FROM   DEPTUS
  EXCEPT
  SELECT DEPTNO,
         DEPTNAME,
         LOCATION
  FROM   DEPT
  ORDER BY DEPTNO
```

```
DEPTNO  DEPTNAME          LOCATION
  601   SALESWESTUS       LOS ANGELES
  600   SALESEASTUS       NEW YORK
********  End of data  ********
```

Set Operators – EXCEPT / MINUS

- Oracle does not allow EXCEPT
- It uses MINUS

```
SELECT DEPTNO, DEPTNAME,LOCATION
FROM   DEPT
MINUS
SELECT DEPTNO, DEPTNAME,LOCATION
FROM   DEPTUS
```

Sets vs Joins

- JOINS are about COMBINING the records from two different tables (or objects) where there is some common connection between the two.
- Example: EMPLOYEE table and SALARY table. Both have employee number and name in common, but, one table focuses on one set of attributes about the employee (like personal information such as phone, address, years of service, etc.) – and – the other focuses on a different set of attributes about the employee (like department, salary, commission, etc).

Ontario and Manitoba

- Ontario

CUSTOMER ID	FIRST NAME	LAST	CITY	PURCHASE	PURCHASE
		NAME		AMOUNT	DATE
432222	Fred	Atkins	Mississauga	444.66	01/12/23
234313	Gordon	Wong	Toronto	225.51	01/02/23
145678	Ken	Stafford	Toronto	95.00	02/03/23
121314	Jill	Wong	Oakville	1,200.00	01/11/23
43234	Kate	Middleton	London	225.49	08/10/23

- Manitoba

CUSTOMER ID	FIRST NAME	LAST	CITY	PURCHASE	PURCHASE
		NAME		AMOUNT	DATE
544444	Ronald	Wing	Winnipeg	9,845.65	12/20/22
555555	Bill	Smith	Brandon	126.33	01/14/23
233333	Gordon	Switzer	Winnipeg	122.99	12/20/22
65234	Jasper	Higgins	Winnipeg	1,345.00	03/03/23
434543	Tabitha	Daniels	Brandon	13.40	02/12/23
143678	Carlos	Santatna	Brandon	150.50	12/22/22

Quebec and all tables

- Quebec

```
CUSTOMER ID  FIRST NAME        LAST             CITY                 PURCHASE   PURCHASE
                               NAME                                  AMOUNT     DATE
  111111     Jacques           Trembley         Montreal               122.33   01/01/23
  122222     Sarah             Singh            Dorval                 233.66   12/31/22
  133333     Rene              Lemieux          Quebec City            336.66   01/16/23
  222222     Sandy             Thomas           Lachine                445.54   12/14/22
```

- Customers from the 3 tables spending more than $200

```
CUSTOMER ID  FULLNAME                         CITY                 PURCHASE    PURCHASE  PROVINCE
                                                                   AMOUNT      DATE
  65234      Jasper Higgins                   Winnipeg             1,345.00    03/03/23  MANITOBA
  544444     Ronald Wing                      Winnipeg             9,845.65    12/20/22  MANITOBA
  234313     Gordon Wong                      Toronto                225.51    01/02/23  ONTARIO
  121314     Jill Wong                        Oakville             1,200.00    01/11/23  ONTARIO
  43234      Kate Middleton                   London                 225.49    08/10/23  ONTARIO
  432222     Fred Atkins                      Mississauga            444.66    01/12/23  ONTARIO
  222222     Sandy Thomas                     Lachine                445.54    12/14/22  QUEBEC
  133333     Rene Lemieux                     Quebec City            336.66    01/16/23  QUEBEC
  122222     Sarah Singh                      Dorval                 233.66    12/31/22  QUEBEC
```

Union for 3 Tables

```
SELECT CUSTID,
       TRIM(FNAME) || ' ' || LNAME FULLNAME,
       CITY, PURCHASE, PDATE,
       'ONTARIO' AS PROVINCE
FROM   ONTARIO
WHERE  PURCHASE > 200
UNION
SELECT CUSTID,
       TRIM(FNAME) || ' ' || LNAME FULLNAME,
       CITY, PURCHASE, PDATE,
       'MANITOBA' AS PROVINCE
FROM   MANITOBA
WHERE  PURCHASE > 200
UNION
SELECT CUSTID,
       TRIM(FNAME) || ' ' || LNAME FULLNAME,
       CITY, PURCHASE, PDATE,
       'QUEBEC' AS PROVINCE
FROM   QUEBEC
WHERE  PURCHASE > 200
ORDER BY PROVINCE
```

Provinces

- Produce a table with all the provinces
- Provinces

```
CUSTOMER ID  FIRST NAME       LAST             CITY                   PURCHASE   PURCHASE   PROV
                              NAME                                    AMOUNT     DATE
  432222     Fred             Atkins           Mississauga              444.66   01/12/23   ONTARIO
  234313     Gordon           Wong             Toronto                  225.51   01/02/23   ONTARIO
  145678     Ken              Stafford         Toronto                   95.00   02/03/23   ONTARIO
  121314     Jill             Wong             Oakville               1,200.00   01/11/23   ONTARIO
  43234      Kate             Middleton        London                   225.49   08/10/23   ONTARIO
  544444     Ronald           Wing             Winnipeg               9,845.65   12/20/22   MANITOBA
  555555     Bill             Smith            Brandon                  126.33   01/14/23   MANITOBA
  233333     Gordon           Switzer          Winnipeg                 122.99   12/20/22   MANITOBA
  65234      Jasper           Higgins          Winnipeg               1,345.00   03/03/23   MANITOBA
  434543     Tabitha          Daniels          Brandon                   13.40   02/12/23   MANITOBA
  143678     Carlos           Santatna         Brandon                  150.50   12/22/22   MANITOBA
  111111     Jacques          Trembley         Montreal                 122.33   01/01/23   QUEBEC
  122222     Sarah            Singh            Dorval                   233.66   12/31/22   QUEBEC
  133333     Rene             Lemieux          Quebec City              336.66   01/16/23   QUEBEC
  222222     Sandy            Thomas           Lachine                  445.54   12/14/22   QUEBEC
```

PROVINCES

```
CREATE TABLE PROVINCES AS (SELECT * FROM ONTARIO) WITH DATA

ALTER TABLE PROVINCES ADD COLUMN PROV CHAR(15)

UPDATE PROVINCES
SET PROV = 'ONTARIO'

INSERT INTO PROVINCES
       (SELECT CUSTID, FNAME,
               LNAME, CITY, PURCHASE,
               PDATE, 'MANITOBA'
        FROM   MANITOBA)

 INSERT INTO PROVINCES
        (SELECT CUSTID, FNAME,
                LNAME, CITY, PURCHASE,
                PDATE, 'QUEBEC'
         FROM   QUEBEC)
```

Excluding the province QUEBEC

- How do you exclude the Quebec rows from the PROVINCES table with a set operator?

```
SELECT * FROM PROVINCES
EXCEPT
SELECT * FROM QUEBEC
```

```
Number of columns not consistent
```

- PROVINCES has an extra column not found in the QUEBEC table.

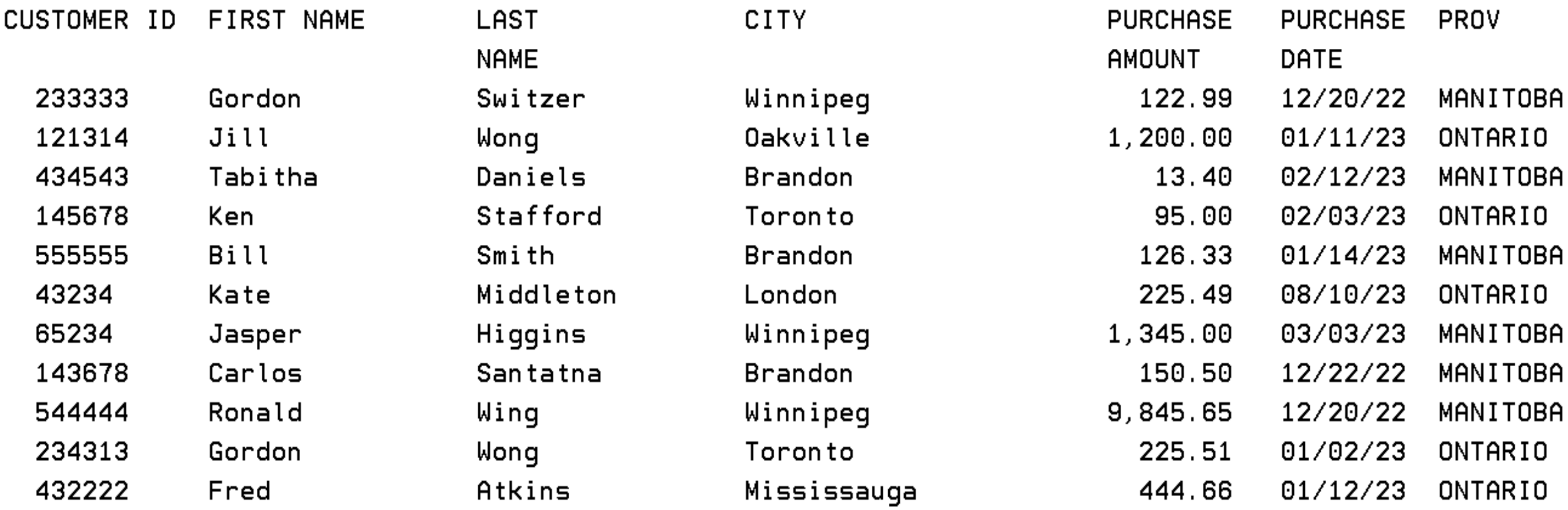
Excluding the province QUEBEC

```
SELECT CUSTID, FNAME,LNAME,CITY,PURCHASE,PDATE
FROM PROVINCES
EXCEPT
SELECT * FROM QUEBEC
```

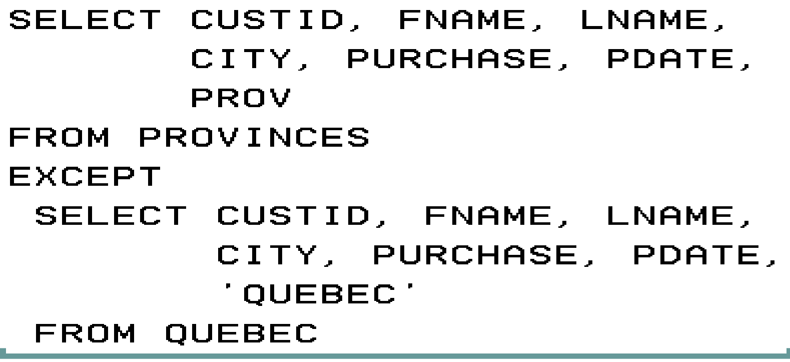
```
CUSTOMER ID  FIRST NAME       LAST              CITY                   PURCHASE   PURCHASE
                              NAME                                     AMOUNT     DATE
  555555     Bill             Smith             Brandon                  126.33   01/14/23
  544444     Ronald           Wing              Winnipeg               9,845.65   12/20/22
  143678     Carlos           Santatna          Brandon                  150.50   12/22/22
  233333     Gordon           Switzer           Winnipeg                 122.99   12/20/22
  145678     Ken              Stafford          Toronto                   95.00   02/03/23
  234313     Gordon           Wong              Toronto                  225.51   01/02/23
  65234      Jasper           Higgins           Winnipeg               1,345.00   03/03/23
  43234      Kate             Middleton         London                   225.49   08/10/23
  432222     Fred             Atkins            Mississauga              444.66   01/12/23
  434543     Tabitha          Daniels           Brandon                   13.40   02/12/23
  121314     Jill             Wong              Oakville               1,200.00   01/11/23
```

Including the province name

- Exclude QUEBEC in PROVINCES and show province?



```
CUSTOMER ID  FIRST NAME    LAST          CITY                PURCHASE  PURCHASE  PROV
                           NAME                              AMOUNT    DATE
  233333     Gordon        Switzer       Winnipeg              122.99  12/20/22  MANITOBA
  121314     Jill          Wong          Oakville            1,200.00  01/11/23  ONTARIO
  434543     Tabitha       Daniels       Brandon                13.40  02/12/23  MANITOBA
  145678     Ken           Stafford      Toronto                95.00  02/03/23  ONTARIO
  555555     Bill          Smith         Brandon               126.33  01/14/23  MANITOBA
  43234      Kate          Middleton     London                225.49  08/10/23  ONTARIO
  65234      Jasper        Higgins       Winnipeg            1,345.00  03/03/23  MANITOBA
  143678     Carlos        Santatna      Brandon               150.50  12/22/22  MANITOBA
  544444     Ronald        Wing          Winnipeg            9,845.65  12/20/22  MANITOBA
  234313     Gordon        Wong          Toronto               225.51  01/02/23  ONTARIO
  432222     Fred          Atkins        Mississauga           444.66  01/12/23  ONTARIO
```



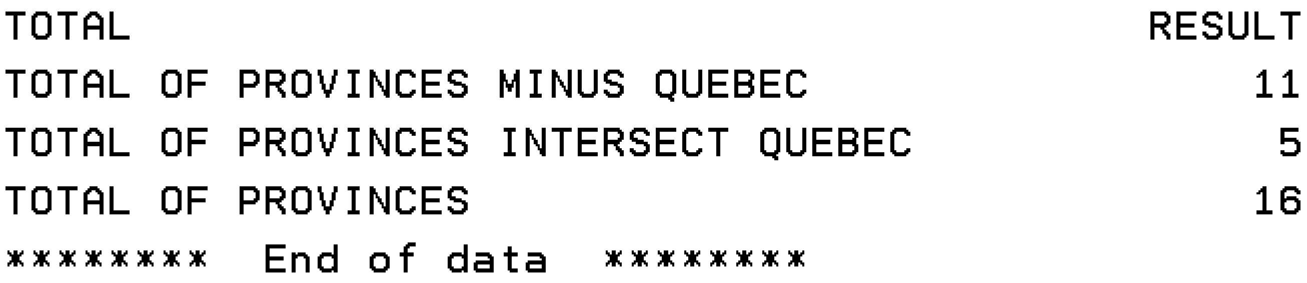
```
SELECT CUSTID, FNAME, LNAME,
       CITY, PURCHASE, PDATE,
       PROV
FROM PROVINCES
EXCEPT
 SELECT CUSTID, FNAME, LNAME,
        CITY, PURCHASE, PDATE,
        'QUEBEC'
 FROM QUEBEC
```

INTERSECT

```
SELECT CUSTID,FNAME,LNAME,CITY, PURCHASE, PDATE
FROM   PROVINCES
INTERSECT
SELECT * FROM QUEBEC
```

```
CUSTOMER ID  FIRST NAME      LAST            CITY                  PURCHASE   PURCHASE
                             NAME                                  AMOUNT     DATE
  111111     Jacques         Trembley        Montreal                122.33   01/01/23
  122222     Sarah           Singh           Dorval                  233.66   12/31/22
  133333     Rene            Lemieux         Quebec City             336.66   01/16/23
  222222     Sandy           Thomas          Lachine                 445.54   12/14/22
  398972     Robert          Trembley        Montreal                144.99   02/02/23
```

- Totals



```
TOTAL                                              RESULT
TOTAL OF PROVINCES MINUS QUEBEC                        11
TOTAL OF PROVINCES INTERSECT QUEBEC                     5
TOTAL OF PROVINCES                                     16
********  End of data  ********
```

```
SELECT 'TOTAL OF PROVINCES MINUS QUEBEC' TOTAL,
        COUNT(*) RESULT
FROM    (SELECT CUSTID, FNAME, LNAME, CITY, PURCHASE, PDATE
         FROM PROVINCES
          EXCEPT
        SELECT * FROM QUEBEC)
UNION
SELECT 'TOTAL OF PROVINCES INTERSECT QUEBEC' TOTAL,
        COUNT(*) RESULT
FROM    (SELECT CUSTID, FNAME, LNAME, CITY, PURCHASE, PDATE
         FROM PROVINCES
          INTERSECT
        SELECT * FROM QUEBEC)
UNION
SELECT 'TOTAL OF PROVINCES' TOTAL, COUNT(*) RESULT
FROM  PROVINCES
```

Using a set operator to solve a Problem

- Thousands of customers and hundreds of employees.
- Are any of our employees listed as customer contacts?

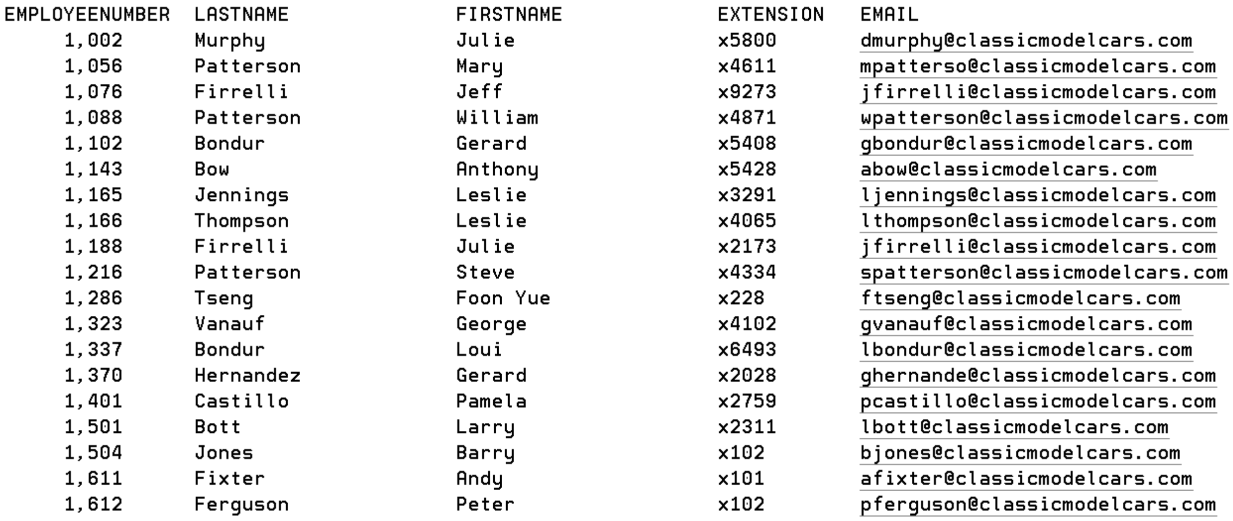
Using a set operator to solve a Problem

- CUSTOMERS

```
CUSTOMERNUMBER  CUSTOMERNAME                      CONTACTLASTNAME            CONTACTFIRSTNAME    PHONE
       103      Atelier graphique                 Schmitt                    Carine              40.32.2555
       112      Signal Gift Stores                King                       Jean                7025551838
       114      Australian Collectors, Co.        Ferguson                   Peter               03 9520 4555
       119      La Rochelle Gifts                 Labrune                    Janine              40.67.8555
       121      Baane Mini Imports                Bergulfsen                 Jonas               07-98 9555
       124      Mini Gifts Distributors Ltd.      Nelson                     Susan               4155551450
       125      Havel "&" Zbyszek Co              Piestrzeniewicz            Zbyszek             (26) 642-7555
       128      Blauer See Auto, Co.              Keitel                     Roland              +49 69 66 90 2555
       129      Mini Wheels Co.                   Murphy                     Julie               6505555787
       131      Land of Toys Inc.                 Lee                        Kwai                2125557818
       141      Euro+ Shopping Channel            Freyre                     Diego               (91) 555 94 44
       144      Volvo Model Replicas, Co          Berglund                   Christina           0921-12 3555
       145      Danish Wholesale Imports          Petersen                   Jytte               31 12 3555
       146      Saveley "&" Henriot, Co.          Saveley                    Mary                78.32.5555
       148      Dragon Souveniers, Ltd.           Natividad                  Eric                +65 221 7555
       151      Muscle Machine Inc                Young                      Jeff                2125557413
       157      Diecast Classics Inc.             Leong                      Kelvin              2155551555
       161      Technics Stores Inc.              Hashimoto                  Juri                6505556809
       166      Handji Gifts"&" Co                Victorino                  Wendy               +65 224 1555
```

Using a set operator to solve a Problem

- EMPLOYEES



EMPLOYEENUMBER	LASTNAME	FIRSTNAME	EXTENSION	EMAIL
1,002	Murphy	Julie	x5800	dmurphy@classicmodelcars.com
1,056	Patterson	Mary	x4611	mpatterso@classicmodelcars.com
1,076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com
1,088	Patterson	William	x4871	wpatterson@classicmodelcars.com
1,102	Bondur	Gerard	x5408	gbondur@classicmodelcars.com
1,143	Bow	Anthony	x5428	abow@classicmodelcars.com
1,165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com
1,166	Thompson	Leslie	x4065	lthompson@classicmodelcars.com
1,188	Firrelli	Julie	x2173	jfirrelli@classicmodelcars.com
1,216	Patterson	Steve	x4334	spatterson@classicmodelcars.com
1,286	Tseng	Foon Yue	x228	ftseng@classicmodelcars.com
1,323	Vanauf	George	x4102	gvanauf@classicmodelcars.com
1,337	Bondur	Loui	x6493	lbondur@classicmodelcars.com
1,370	Hernandez	Gerard	x2028	ghernande@classicmodelcars.com
1,401	Castillo	Pamela	x2759	pcastillo@classicmodelcars.com
1,501	Bott	Larry	x2311	lbott@classicmodelcars.com
1,504	Jones	Barry	x102	bjones@classicmodelcars.com
1,611	Fixter	Andy	x101	afixter@classicmodelcars.com
1,612	Ferguson	Peter	x102	pferguson@classicmodelcars.com

Using a set operator to solve a Problem

- COMMON NAMES?

```
SELECT CONTACTFIRSTNAME,
       CONTACTLASTNAME
FROM   CUSTOMERS
INTERSECT
SELECT FIRSTNAME,
       LASTNAME
FROM EMPLOYEES
```

```
CONTACTFIRSTNAME           CONTACTLASTNAME
Julie                      Murphy
Peter                      Ferguson
```

Phone the employee and ask

- Include the employee’s extension

```
FIRSTNAME             LASTNAME              EXTENSION
Julie                 Murphy                x5800
Peter                 Ferguson              x102
********  End of data  ********
```

```
SELECT FIRSTNAME,
       LASTNAME,
       EXTENSION
FROM   EMPLOYEES
WHERE (FIRSTNAME,LASTNAME) IN
      (SELECT CONTACTFIRSTNAME,
              CONTACTLASTNAME
       FROM   CUSTOMERS
INTERSECT
SELECT FIRSTNAME,
       LASTNAME
FROM EMPLOYEES )
```

Set Operators – Additional Guidelines

- You must match the data type when columns do not exist in one or the other table
- You can use various functions like CAST, TO_CHAR or any other conversion function to ensure data types match
- SELECT department_id,
- TO_NUMBER (null) as location,
- hire_date
- FROM employees
- UNION
- SELECT department_id,
- location_id,
- TO_DATE (null)
- FROM departments;

Set Operators – Additional Guidelines

- when columns do not exist in one or the other table
- SELECT employee_id,
- job_id, salary
- FROM employees
- UNION
- SELECT employee_id,
- Job_id,
- 0
- FROM job_history;

Set Operators – Additional Guidelines

- The expressions in the SELECT lists must match in number.
- If you select 3 columns in A, then must have 3 columns in B
- The data type of each column in the second query must match the data type of its corresponding column in the first query.
- Parentheses can be used to alter the sequence of execution.
- ORDER BY clause can appear only at the very end of the statement.
- Duplicate rows are automatically eliminated except in UNION ALL.
- Column names from the first query are the ones that appear in the result.
- The output is sorted in ascending order by default except in UNION ALL

Lab 4

- Lab 4 is posted.
- Review for Midterm Test is next week
- Assignment 1 is due in week 6
- Midterm test is in week 6