XP2
SQL TRAINING

Mike Mohr
Data Scientist
United Solutions Company
mmohr@unitedsolutions.coop

United Solutions

fiserv.

INTRODUCTION

Purpose of the class

SQL at United Solutions

Need help?

CS@UnitedSolutions.coop

850-942-9186 ext 4200

United Solutions

fiserv.



United Solutions | fiserv.

AGENDA

- Scalar Functions
  - Built-in functions that return a single value
- Aggregate Functions
  - Performs calculations and returns a single value
- Common Table Expressions
  - Temporary table that can be referenced in a query
- Window Functions
  - Perform calculations across rows related to a single row



SCALAR FUNCTIONS

- **COALESCE**
  - **Returns 1st non-null value from list**



United Solutions | fiserv.

SCALAR FUNCTIONS

- **COALESCE**
  - **Returns 1st non-null value from list**

```
SELECT SHARE.MEMBER_NBR, SHARE.SHARE_NBR
FROM DB2INST1.SHARE SHARE
WHERE SHARE.CLOSED = '0'
```



SCALAR FUNCTIONS

- **COALESCE**
  - **Returns 1st non-null value from list**

```
SELECT SHARE.MEMBER_NBR,
SHARE.SHARE_NBR
FROM DB2INST1.SHARE SHARE
WHERE SHARE.CLOSED = '0'
```

```
SELECT SHARE.MEMBER_NBR,
SHARE.SHARE_NBR
FROM DB2INST1.SHARE SHARE
WHERE COALESCE(SHARE.CLOSED, '0') = '0'
```



SCALAR FUNCTIONS

DATE FUNCTIONS

- **CURRENT_DATE – current system date and time**



SCALAR FUNCTIONS

DATE FUNCTIONS

- **CURRENT_DATE – current system date**

```
SELECT
SHARE.MEMBER_NBR,
CURRENT_DATE
FROM DB2INST1.SHARE
SHARE
```

123 MEMBER_NBR	CURRENT_DATE
1	2026-06-22
2	2026-06-22
3	2026-06-22
4	2026-06-22
5	2026-06-22
6	2026-06-22
7	2026-06-22
8	2026-06-22
9	2026-06-22
10	2026-06-22
11	2026-06-22
12	2026-06-22
13	2026-06-22
14	2026-06-22



SCALAR FUNCTIONS

DATE FUNCTIONS

- CURRENT_DATE – current system date

```
SELECT SHARE.MEMBER_NBR, CURRENT_DATE
FROM DB2INST1.SHARE SHARE
```

- Add/subtract calendar periods to/from date
  - (FIELD1 + NUMBER DatePart)



SCALAR FUNCTIONS

DATE FUNCTIONS

- CURRENT_DATE – current system date

```
SELECT SHARE.MEMBER_NBR, CURRENT_DATE
FROM DB2INST1.SHARE SHARE
```

- Add/subtract calendar periods to/from date
  - (FIELD1 + NUMBER DatePart)

```
SELECT SHARE.MEMBER_NBR, SHARE.SHARE_NBR,
SHARE.OPEN_DATE, CURRENT_DATE AS
CURRENT_DATE
, (SHARE.OPEN_DATE + 30 DAYS) AS THIRTY_DAYS
FROM DB2INST1.SHARE SHARE
```



United Solutions | fiserv.

SCALAR FUNCTIONS

123 MEMBER_NBR	123 SHARE_NBR	CURRENT_DATE	OPEN_DATE	THIRTY_DAYS
1	1	2026-06-22	1998-05-06	1998-06-05
1	5	2026-06-22	2007-04-01	2007-05-01
1	7	2026-06-22	1998-05-06	1998-06-05
1	11	2026-06-22	1998-05-06	1998-06-05
1	12	2026-06-22	2005-09-30	2005-10-30
1	20	2026-06-22	1998-05-06	1998-06-05
2	1	2026-06-22	1998-05-06	1998-06-05
2	7	2026-06-22	1998-05-06	1998-06-05
2	11	2026-06-22	1998-05-06	1998-06-05
2	18	2026-06-22	2006-02-16	2006-03-18
2	20	2026-06-22	1998-05-06	1998-06-05
3	1	2026-06-22	1998-07-01	1998-07-31
3	2	2026-06-22	1999-04-24	1999-05-24
3	20	2026-06-22	1999-04-24	1999-05-24
4	1	2026-06-22	1998-07-01	1998-07-31
4	2	2026-06-22	1999-04-24	1999-05-24
4	20	2026-06-22	1999-04-24	1999-05-24
5	1	2026-06-22	1998-07-01	1998-07-31
5	2	2026-06-22	1999-04-24	1999-05-24
6	1	2026-06-22	1999-04-24	1999-05-24
6	2	2026-06-22	1999-04-24	1999-05-24



United Solutions | fiserv.

SCALAR FUNCTIONS

DATE FUNCTIONS

- Add/subtract date fields from each other
- Days_Between

```
SELECT SHARE.MEMBER_NBR, SHARE.SHARE_NBR,
SHARE.OPEN_DATE, YEAR(CURRENT_DATE) -
YEAR(SHARE.OPEN_DATE) AS YEARS_OPEN,
DAYS_BETWEEN(CURRENT_DATE, SHARE.OPEN_DATE)
AS DAYS_OPEN

FROM DB2INST1.SHARE SHARE

WHERE COALESCE(SHARE.CLOSED, '0') = '0'
```



United Solutions | fiserv.

SCALAR FUNCTIONS

DATE FUNCTIONS

MEMBER_NBR	SHARE_NBR	OPEN_DATE	YEARS_OPEN	DAYS_OPEN
1	1	1998-05-06	28	10274
1	5	2007-04-01	19	7022
1	7	1998-05-06	28	10274
1	11	1998-05-06	28	10274
1	12	2005-09-30	21	7570
1	20	1998-05-06	28	10274
2	1	1998-05-06	28	10274
2	7	1998-05-06	28	10274
2	11	1998-05-06	28	10274
2	18	2006-02-16	20	7431
2	20	1998-05-06	28	10274
3	1	1998-07-01	28	10218
3	2	1999-04-24	27	9921
3	20	1999-04-24	27	9921
4	1	1998-07-01	28	10218
4	2	1999-04-24	27	9921
4	20	1999-04-24	27	9921
5	1	1998-07-01	28	10218
5	2	1999-04-24	27	9921
6	1	1999-04-24	27	9921
6	2	1999-04-24	27	9921



United Solutions | fiserv.

SCALAR FUNCTIONS

DATE FUNCTIONS

- **YEAR/MONTH/DAY – Extract the date part from a date**



SCALAR FUNCTIONS

DATE FUNCTIONS

- YEAR/MONTH/DAY – Extract the date part from a date
- DATEPART(DATE)
  - YEAR(DATE)
  - MONTH(DATE)
  - DAY(DATE)

MEMBER_NBR	SHARE_NBR	OPEN_DATE	Y	M	D
1	1	1998-05-06	1998	5	6
1	5	2007-04-01	2007	4	1
1	7	1998-05-06	1998	5	6
1	11	1998-05-06	1998	5	6
1	12	2005-09-30	2005	9	30
1	20	1998-05-06	1998	5	6
2	1	1998-05-06	1998	5	6
2	7	1998-05-06	1998	5	6
2	11	1998-05-06	1998	5	6
2	18	2006-02-16	2006	2	16
2	20	1998-05-06	1998	5	6
3	1	1998-07-01	1998	7	1
3	2	1999-04-24	1999	4	24
3	20	1999-04-24	1999	4	24
4	1	1998-07-01	1998	7	1



SCALAR FUNCTIONS

- **LEFT/RIGHT**
  - **Returns leftmost or rightmost characters in a field**
  - **LEFT/RIGHT(FIELD,LENGTH)**

```
SELECT INDIVIDUAL.TIN,
RIGHT(INDIVIDUAL.TIN,4)
FROM DB2INST1.INDIVIDUAL
INDIVIDUAL
```



SCALAR FUNCTIONS

- **LEFT/RIGHT**

123 TIN	AZ LAST_4
12300567	0567
12345623	5623
12345662	5662
15975312	5312
19441957	1957
25464444	4444
25500663	0663
25894896	4896
40360149	0149
45627952	7952
58957489	7489
65356515	6515
70070707	0707
88880088	0088
90009876	9876
100000001	0001
100000002	0002
100000003	0003
100000004	0004
100000007	0007
100000008	0008
100000010	0010



SCALAR FUNCTIONS

- **LPAD/RPAD**
  - **Adds leading or trailing characters to a field to a set length**
  - **LPAD/RPAD(FIELD, TOTAL LENGTH, 'CHARACTER')**



SCALAR FUNCTIONS

- **LPAD/RPAD**
  - **Adds leading or trailing characters to a field to a set length**
  - **LPAD/RPAD(FIELD, TOTAL LENGTH, 'CHARACTER')**

```
SELECT DISTINCT ACCOUNT.MEMBER_NBR,
LPAD(ACCOUNT.MEMBER_NBR,10,'0') AS
LONG_MBR
FROM DB2INST1.ACCOUNT ACCOUNT
```



United Solutions | fiserv.

SCALAR FUNCTIONS

123 MEMBER_NBR	A-Z LONG_MBR
1	0000000001
2	0000000002
3	0000000003
4	0000000004
5	0000000005
6	0000000006
7	0000000007
8	0000000008
9	0000000009
10	0000000010
11	0000000011
12	0000000012
13	0000000013
14	0000000014
15	0000000015
101	0000000101
102	0000000102
103	0000000103
104	0000000104
105	0000000105
106	0000000106
107	0000000107
108	0000000108



United Solutions | fiserv.

AGGREGATE FUNCTIONS

MIN	lowest value
MAX	highest value
SUM	total
AVG	average
COUNT	number of items in column
COUNT (*)	number of rows in table



AGGREGATE FUNCTIONS

MIN	lowest value
MAX	highest value
SUM	total
AVG	average
COUNT	number of items in column
COUNT (*)	number of rows in table

```
SELECT
COUNT(MEMBERSHIP.MEMBER_NBR)
FROM DB2INST1.MEMBERSHIP
MEMBERSHIP
```



United Solutions | fiserv.

AGGREGATE FUNCTIONS

MIN	lowest value
MAX	highest value
SUM	total
AVG	average
COUNT	number of items in column
COUNT (*)	number of rows in table

```
SELECT SUM(SHARE.BALANCE)
FROM DB2INST1.SHARE SHARE
```



AGGREGATE FUNCTIONS

- **GROUP BY – Subtotaling function**
  - **Used with Aggregate Functions**



United Solutions | fiserv.

AGGREGATE FUNCTIONS

- **GROUP BY – Subtotaling function**
  - **Used with Aggregate Functions**

```
SELECT SUM(SHARE.BALANCE)
FROM DB2INST1.SHARE SHARE
```



United Solutions | fiserv.

AGGREGATE FUNCTIONS

- **GROUP BY – Subtotaling function**
  - **Used with Aggregate Functions**

```
SELECT SHARE.MEMBER_NBR, SUM(SHARE.BALANCE)
        FROM DB2INST1.SHARE SHARE
        GROUP BY SHARE.MEMBER_NBR
```



AGGREGATE FUNCTIONS

123 MEMBER_NBR	123 TOTAL_BAL
1	38479.99
2	623635.28
3	1383790.06
4	1399528.99
5	10961.51
6	141.91
7	1400114.96
8	1540.15
9	34.99

- **GROUP BY – Subtotaling function**
  - **Used with Aggregate Functions**

```
SELECT SHARE.MEMBER_NBR,
SUM(SHARE.BALANCE)
        FROM DB2INST1.SHARE SHARE
        GROUP BY SHARE.MEMBER_NBR
```

All non-aggregate items in the SELECT clause must be in the GROUP BY clause



AGGREGATE FUNCTIONS

- **HAVING – Filters grouped data**



United Solutions | fiserv.

AGGREGATE FUNCTIONS

- **HAVING – Filters grouped data**

```
SELECT SHARE.MEMBER_NBR,
SUM(SHARE.BALANCE)
FROM DB2INST1.SHARE SHARE
GROUP BY SHARE.MEMBER_NBR
```



AGGREGATE FUNCTIONS

- **HAVING – Filters grouped data**

```
SELECT SHARE.MEMBER_NBR, SUM(SHARE.BALANCE)
FROM DB2INST1.SHARE SHARE
GROUP BY SHARE.MEMBER_NBR
HAVING SUM(SHARE.BALANCE) >10000
```



AGGREGATE FUNCTIONS

- **HAVING – Filters grouped data**

```
SELECT SHARE.MEMBER_NBR, SUM(SHARE.BALANCE)
FROM DB2INST1.SHARE SHARE
WHERE SHARE.BALANCE >1000
GROUP BY SHARE.MEMBER_NBR
HAVING SUM(SHARE.BALANCE) >10000
```



AGGREGATE FUNCTIONS

- **HAVING – Filters grouped data**

123 MEMBER_NBR	123 HIGH_BAL
1	38382.69
2	622628.03
3	1383790.06
4	1398528.99
5	10860.38
7	1399834.94
14	699973.46
15	7000562.76
710	19772.06
1119	41436.57
2000	10710.55
2001	29269.23
2003	265245.14
2004	11131.49
2005	935136.12
2007	175190.11
2008	83630.76
2009	146525.86
2010	59037.48
2011	51563.64



United Solutions | fiserv.

COMMON TABLE EXPRESSIONS

- CTE – temporary tables
- Created before the main query
  - WITH *CTE* AS (CTE…)



United Solutions | fiserv.

COMMON TABLE EXPRESSIONS

```
WITH HIGH_BAL AS
 (SELECT SHARE.MEMBER_NBR, SUM(SHARE.BALANCE) AS BAL
        FROM DB2INST1.SHARE SHARE
        WHERE SHARE.BALANCE >1000
        GROUP BY SHARE.MEMBER_NBR
        HAVING SUM(SHARE.BALANCE) >10000)
```



COMMON TABLE EXPRESSIONS

```
WITH HIGH_BAL AS
 (SELECT SHARE.MEMBER_NBR, SUM(SHARE.BALANCE) AS BAL
        FROM DB2INST1.SHARE SHARE
        WHERE SHARE.BALANCE >1000
        GROUP BY SHARE.MEMBER_NBR
        HAVING SUM(SHARE.BALANCE) >10000)
, DLQ AS (SELECT DISTINCT AGR_LOANDELQ.MEMBER_NBR
        FROM DB2INST1.AGR_LOANDELQ AGR_LOANDELQ
        WHERE AGR_LOANDELQ.DAYSDEL > 30)
```



United Solutions | fiserv.

COMMON TABLE EXPRESSIONS

```
WITH HIGH_BAL AS
 (SELECT SHARE.MEMBER_NBR, SUM(SHARE.BALANCE) AS BAL
                FROM DB2INST1.SHARE SHARE
                WHERE SHARE.BALANCE >1000
                GROUP BY SHARE.MEMBER_NBR
                HAVING SUM(SHARE.BALANCE) >10000)
, DLQ AS (SELECT DISTINCT AGR_LOANDELQ.MEMBER_NBR
                FROM DB2INST1.AGR_LOANDELQ AGR_LOANDELQ
                WHERE AGR_LOANDELQ.DAYSDEL > 30)
SELECT M.MEMBER_NBR, HB.BAL
FROM DB2INST1.MEMBERSHIP M
JOIN HIGH_BAL HB
ON M.MEMBER_NBR = HB.MEMBER_NBR
LEFT JOIN DLQ DLQ
ON M.MEMBER_NBR = DLQ.MEMBER_NBR
WHERE DLQ.MEMBER_NBR IS NULL
```



United Solutions | fiserv.

COMMON TABLE EXPRESSIONS

MEMBER_NBR	BAL
1	38382.69
2	622628.03
3	1383790.06
4	1398528.99
5	10860.38
7	1399834.94
14	699973.46
15	7000562.76
10001	10328.37
2000	10710.55
2001	29269.23
2003	265245.14
2004	11131.49
2005	935136.12
2007	175190.11
2008	83630.76
2009	146525.86
2010	59037.48
2011	51563.64
2012	81683.27
2013	43972.64
2014	77869.88
2015	45686.95



United Solutions | fiserv.

WINDOW FUNCTIONS

- **Calculate from multiple rows *for each row separately***
  - **Aggregate functions calculate *for all rows***



WINDOW FUNCTIONS

- **Calculate from multiple rows *for each row separately***
  - **Aggregate functions calculate *for all rows***

Rank

- **Row_number ()**
- **Rank ()**
- **Dense_Rank ()**
- **nTile(n)**



United Solutions | fiserv.

WINDOW FUNCTIONS

- **Calculate from multiple rows *for each row separately***
  - **Aggregate functions calculate *for all rows***

Rank

- **Row_number ()**
- **Rank ()**
- **Dense_Rank ()**
- **nTile(n)**

Value

- **Lag ()**
- **Lead ()**
- **First_Value ()**
- **Last_Value ()**



United Solutions | fiserv.

WINDOW FUNCTIONS

- **Calculate from multiple rows *for each row separately***
  - **Aggregate functions calculate *for all rows***

Rank
- **Row_number ()**
- **Rank ()**
- **Dense_Rank ()**
- **nTile(n)**

Value
- **Lag ()**
- **Lead ()**
- **First_Value ()**
- **Last_Value ()**

Aggregate
- **Sum ()**
- **Avg ()**
- **Count ()**



United Solutions | fiserv.

WINDOW FUNCTIONS

```
SELECT
        column_name,
        FUNCTION() OVER (
        PARTITION BY group_column
        ORDER BY sort_column
        ROWS BETWEEN 1 PRECEDING AND CURRENT ROW
        ) AS alias_name

FROM table_name
```



United Solutions | fiserv.

WINDOW FUNCTIONS

```
SELECT EMAIL.INDIVIDUAL_ID
, EMAIL.EMAIL_ADDRESS
, ROW_NUMBER() OVER (PARTITION BY
EMAIL.INDIVIDUAL_ID ORDER BY EMAIL.SEQ DESC) AS
RN

FROM DB2INST1.EMAIL EMAIL

WHERE COALESCE(EMAIL.IS_PRIMARY, '1') = 1
```



United Solutions | fiserv.

WINDOW FUNCTIONS

123 INDIVIDUAL_ID	A-Z EMAIL_ADDRESS	123 RN
172	JANPA@MSN.COM	1
177	www@aaa.com	1
177	williamw@xxx.com	2
185	jaymackay@hotmail.com	1
186	pjillson@comcast.net	1
187	gfranklin@adelphia.net	1
188	jj49@interwest.com	1
196	barbd@adelphia.net	1
205	kaseywinger@bellsouth.com	1
206	cphillips@hotmail.com	1
222	melr@sbc.com	1
228	cwinwood@hotmail.com	1
234	mnmnm@verizon.net	1
248	adell.smith@gmail.com	1
250	H.Henshaw@gmail.com	1
266	MA@MA.com	1
330	tomplummer@hotmail.com	1
392	esmithson@yahoo.com	1
393	ericsbikes@aol.com	1
394	joanne@azmedicalassociates.com	1
394	bonnie@azmedicalassociates.com	2
396	john@jjdaycare.com	1
397	bill@rockytopads.com	1
398	BILL@SWAPC.COM	1



United Solutions | fiserv.

WINDOW FUNCTIONS

```
WITH EMAIL AS (SELECT EMAIL.INDIVIDUAL_ID
, EMAIL.EMAIL_ADDRESS
, ROW_NUMBER() OVER (PARTITION BY EMAIL.INDIVIDUAL_ID ORDER BY EMAIL.SEQ DESC) AS RN
FROM DB2INST1.EMAIL EMAIL
WHERE COALESCE(EMAIL.IS_PRIMARY, '1') = ‘1’)

SELECT DISTINCT I.FIRST_NAME
, I.LAST_NAME
, E.EMAIL_ADDRESS AS EMAIL
, SUM(S.BALANCE) AS TOTAL_BAL
FROM DB2INST1.MEMBERSHIPPARTICIPANT MP
JOIN DB2INST1.SHARE S
ON  MP.MEMBER_NBR = S.MEMBER_NBR
                AND COALESCE(S.CLOSED,'0') = '0'
JOIN DB2INST1.INDIVIDUAL I
ON MP.INDIVIDUAL_ID = I.INDIVIDUAL_ID
JOIN EMAIL E
ON I.INDIVIDUAL_ID = E.INDIVIDUAL_ID
                 AND E.RN = 1
 WHERE MP.PARTICIPATION_TYPE = '101’
GROUP BY I.FIRST_NAME
                , I.LAST_NAME
                , E.EMAIL_ADDRESS
```



WINDOW FUNCTIONS

A-Z FIRST_NAME	A-Z LAST_NAME	A-Z EMAIL	123 TOTAL_BAL
KEITH	APPLEBUTTER	kemery@email.com	427.10
FRANK	ASHTON	kme@kme.Kme.com	8099.75
DINA	AUTH	DINAA@TRIVALLEYSERVICE.COM	1519.61
KEITH	AVERY	kme@kme.com	13807.80
KEITH	AXEL	KME@KME.COM	5287.81
Derek	Barker	DBarker@unitedsolutions.coop	4233.47
WILSON	BALL	SCOTT@HOME.COM	11297.80
ADAM	BARCLAY	aaron.barclay@fiserv.com	371.60
DENNIS	BAUER	scott@gmail.com	29789.94
JACK	BAUER	jackb@ctu.com	146960.40
STEVEN	BAUER	SMBAUER@AOL.COM	65711.73
JOE	BAXTER	JOE@MSN.COM	63064.88
Drew	Bender	drew@yahoo.com	1106.29
WILLIAM	BELANGER	WILLB@YAHOO.COM	7.15
JENNIFER	BELL	MYOWN NUMBER @GMAIL.COM	3569.28
PEGGY	BELL	dixiegirl5775@yahoo.com	2366.08
WILLIAM	BENNETT	SCOTT@GMAIL.COM	11247.12
DUSTY	BENSON	DUSTY@ATT.NET	1109.73
JUSTIN	BERBER	justin@aol.com	8880.66
MARY	BERBER	mary@aol.com	2395.25
PETE	BERMAN	scott@home.com	11350.08
STEVE	BERMANN	scott@gmail.com	11342.24



Please email or call and we will be happy to answer any questions or provide assistance with SQL Report creation.

Mike Mohr
Data Scientist
United Solutions Company
mmohr@unitedsolutions.coop

WWW.XP2COMMUNITY.COM

Log in and begin today!

United Solutions

fiserv.

Thank You

We look forward to hosting you again!

June 24th 2027

United Solutions

fiserv.