

Analytics Training

Introduction

Welcome to the Analytics Training for the BC Perinatal Data Registry (PDR).

The Analytics Training package is a self-learning tool for health authority and hospital staff who use locally available data from the PDR. The training includes Analytics Training Exercises that are designed to assist staff in generating accurate and reliable information.

To assist learners in working with these Analytics Training Exercises, we have created a Sample Data Set.* This data set has been generated only for training purposes and does not contain original data. An answer key is available to check your results.

It is expected that the learner has sufficient knowledge of the software they use when applying the Analytics Training Exercises.

Currently, there are two levels of samples available, which should be done in order for optimal learning:

Level 1: Introduction to the Basics

1. Count Delivery Episodes Between Two Dates
2. Count Postpartum Episodes of Care Between Two Dates
3. Count Birth Episodes Between Two Dates
4. Count Babies by Live Birth or Stillbirth Between Two Dates
5. Count Baby Transfer/Readmission Episodes Between Two Dates
6. Calculate Parity

Level 2: Beyond the Basics

1. Mother Delivery Method
2. Mother Delivery Method 2
3. Number of Births – Mother, Baby
4. Labour Induction
5. Vaginal Birth After Cesarean
6. Gestational Age Calculation

PSBC can provide clarification about the Analytics Training Exercises and working with PDR data, but we are unable to offer technical assistance with any analytic software applications. If you have a question and/or comment about these examples, please email us at psbc@phsa.ca with the subject line “Analytics Training.”

*** Disclaimer and Limitation of Liability**

Any information created by us is believed to be reliable when posted. However we do not guarantee or warrant the quality, accuracy, completeness, timeliness, appropriateness or suitability of the information provided. Reliance on any information appearing on the website is at your own risk. We expressly disclaim all warranties, representations and conditions regarding use of the website or the information. We assume no obligation to update the information or advise on further developments concerning topics mentioned. The information is supplied “as is” and may contain errors. The information may be changed from time to time without notice. Certain portions of the information may have been contributed by other persons and should not be assumed to have been reviewed or endorsed by us. The mention of specific products or services on this website does not constitute or imply a recommendation or endorsement by us, unless explicitly stated.

The PHSA and its content providers including without limitation the physicians and pharmacists who contribute to the information on this website shall have no liability whether direct, indirect, consequential, contingent, special or incidental related to or arising out of or in connection with the use of this website or your reliance on the information, whether in an action of contract, negligence or other tortious action. Anyone using this information does so at his or her own risk, and by using such information agrees to indemnify the PHSA and its content providers from any and all liability, loss, injury, damages, costs and expenses (including legal fees and expenses) arising from such person's use of the information on this website and any information on websites linked from this website.

NOTES FOR WORKING WITH PDR DATA

PDR Data Structure

The data are organized and stored within specific topic tables. The tables contain columns (variables) and rows (records). Multiple tables containing information on the same mother/baby are linked using a common variable (e.g. [mother_id](#), [baby_id](#), or [patient_id](#)).

All tables referred to as the source for variables are from the PDR and use the following notation:

TABLES: formatted in upper case, e.g. MOTHER_ADMISSION.

VARIABLES: formatted in blue lower case, e.g. [mother_id](#), [multiple_birth_count](#).

For further details regarding the scope, field descriptions, field options, etc. please refer to the applicable version(s) of the [BC Perinatal Data Registry Reference Manual](#).

Joining (Merging) Data Tables

We have designed these examples to be software independent but have chosen to explain the concept of joining (or merging) data tables using SQL language. We will illustrate the concepts by doing different joins on two example tables, MOTHER_ADMISSION and PROCEDURES_PERFORMED.

MOTHER_ADMISSION (table 1)

mother_id	screen_source
1591677	PP
1593206	PP
1594512	DL
1595739	DL
1595752	PP
1595754	DL
1599085	DL
1599089	DL

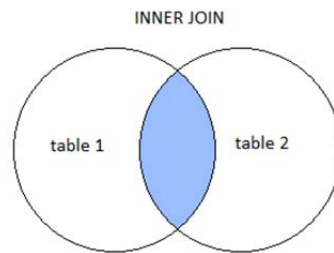
PROCEDURES_PERFORMED (table 2)

patient_id	procedure_code
1591677	1RM87CAAE
1591677	5PC80JP
1602588	5MD50AA
1602600	5PC80JQ
1602617	5MD50AA
1595754	5PC91GA
1603918	5MD60AA
1605509	5AC30HAI2

There are four main types of joins available in SQL:

1. **INNER JOIN:** returns rows when there is a match in both tables.

The INNER JOIN creates a new result table by combining column values of two tables (table 1 and table 2) based upon the common identifier(s). The query compares each row of table 1 with each row of table 2 to find all pairs of rows which have records in both tables. If an identifier is found in both tables, column values for each matched pair of rows of table 1 and table 2 are combined into a result row.



Inner Join Example:

New Table	Table	Field	Action/Formula
INNER_JOIN	MOTHER_ADMISSION	mother_id	MOTHER_ADMISSION
	PROCEDURES_PERFORMED	patient_id	inner join PROCEDURES_PERFORMED

Records Color: exist in both tables, exist only in left table, exist only in right table

MOTHER_ADMISSION (table 1)

mother_id	screen_source
1591677	PP
1593206	PP
1594512	DL
1595739	DL
1595752	PP
1595754	DL
1599085	DL
1599089	DL

PROCEDURES_PERFORMED (table 2)

patient_id	procedure_code
1591677	1RM87CAAE
1591677	5PC80JP
1602588	5MD50AA
1602600	5PC80JQ
1602617	5MD50AA
1595754	5PC91GA
1603918	5MD60AA
1605509	5AC30HA12



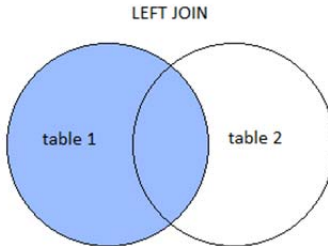
INNER_JOIN

mother_id	screen_source	patient_id	procedure_code
1591677	PP	1591677	1RM87CAAE
1591677	PP	1591677	5PC80JP
1595754	DL	1595754	5PC91GA

2. **LEFT JOIN:** returns all rows from the left table, even if there are no matches in the right table.

If an identifier is found in the left table (table 1) but not the right table (table 2), the join will still return a row in the result, but with NULL in each column from right table.

This means that a left join returns all the values from the left table, plus matched values from the right table or NULL in case of no matching join predicate.



Left Join Example:

New Table	Table	Field	Action/Formula
LEFT_JOIN	MOTHER_ADMISSION	mother_id	MOTHER_ADMISSION
	PROCEDURES_PERFORMED	patient_id	left join PROCEDURES_PERFORMED

Records Color: exist in both tables, exist only in left table, exist only in right table

MOTHER_ADMISSION (table 1)

mother_id	screen_source
1591677	PP
1593206	PP
1594512	DL
1595739	DL
1595752	PP
1595754	DL
1599085	DL
1599089	DL

PROCEDURES_PERFORMED (table 2)

patient_id	procedure_code
1591677	1RM87CAAE
1591677	5PC80JP
1602588	5MD50AA
1602600	5PC80JQ
1602617	5MD50AA
1595754	5PC91GA
1603918	5MD60AA
1605509	5AC30HA12



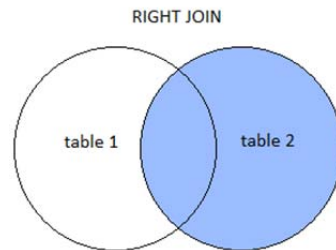
LEFT_JOIN

mother_id	screen_source	patient_id	procedure_code
1591677	PP	1591677	1RM87CAAE
1591677	PP	1591677	5PC80JP
1595754	DL	1595754	5PC91GA
1593206	PP	NULL	NULL
1594512	DL	NULL	NULL
1595739	DL	NULL	NULL
1595752	PP	NULL	NULL
1599085	DL	NULL	NULL
1599089	DL	NULL	NULL

3. **RIGHT JOIN:** returns all rows from the right table, even if there are no matches in the left table.

If an identifier is found in the right table (table 2) but not the left table (table 1), the join will still return a row in the result, but with NULL in each column from left table.

This means that a right join returns all the values from the right table, plus matched values from the left table or NULL in case of no matching join predicate.



Right Join Example:

New Table	Table	Field	Action/Formula
RIGHT_JOIN	MOTHER_ADMISSION	mother_id	MOTHER_ADMISSION
	PROCEDURES_PERFORMED	patient_id	right join PROCEDURES_PERFORMED

Records Color: exist in both tables, exist only in left table, exist only in right table

MOTHER_ADMISSION (table 1)

mother_id	screen_source
1591677	PP
1593206	PP
1594512	DL
1595739	DL
1595752	PP
1595754	DL
1599085	DL
1599089	DL

PROCEDURES_PERFORMED (table 2)

patient_id	procedure_code
1591677	1RM87CAAE
1591677	5PC80JP
1602588	5MD50AA
1602600	5PC80JQ
1602617	5MD50AA
1595754	5PC91GA
1603918	5MD60AA
1605509	5AC30HAI2

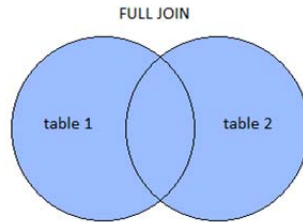


RIGHT_JOIN

mother_id	screen_source	patient_id	procedure_code
1591677	PP	1591677	1RM87CAAE
1591677	PP	1591677	5PC80JP
1595754	DL	1595754	5PC91GA
NULL	NULL	1602588	5MD50AA
NULL	NULL	1602600	5PC80JQ
NULL	NULL	1602617	5MD50AA
NULL	NULL	1603918	5MD60AA
NULL	NULL	1605509	5AC30HAI2

4. **FULL JOIN**: returns rows when there is a match in either one of the tables.

The SQL **FULL JOIN** combines the results of both left and right outer joins. The joined table will contain all records from both tables and fill in NULLs for missing matches on either side.



Full Join Example:

New Table	Table	Field	Action/Formula
FULL_JOIN	MOTHER_ADMISSION	mother_id	MOTHER_ADMISSION
	PROCEDURES_PERFORMED	patient_id	full join PROCEDURES_PERFORMED

Records Color: **exist in both tables**, exist only in left table, exist only in right table

MOTHER_ADMISSION (table 1)

mother_id	screen_source
1591677	PP
1593206	PP
1594512	DL
1595739	DL
1595752	PP
1595754	DL
1599085	DL
1599089	DL

PROCEDURES_PERFORMED (table 2)

patient_id	procedure_code
1591677	1RM87CAAE
1591677	5PC80JP
1602588	5MD50AA
1602600	5PC80JQ
1602617	5MD50AA
1595754	5PC91GA
1603918	5MD60AA
1605509	5AC30HA2



FULL_JOIN

mother_id	screen_source	patient_id	procedure_code
1591677	PP	1591677	1RM87CAAE
1591677	PP	1591677	5PC80JP
1595754	DL	1595754	5PC91GA
1593206	PP	NULL	NULL
1594512	DL	NULL	NULL
1595739	DL	NULL	NULL
1595752	PP	NULL	NULL
1599085	DL	NULL	NULL
1599089	DL	NULL	NULL
NULL	NULL	1602588	5MD50AA
NULL	NULL	1602600	5PC80JQ
NULL	NULL	1602617	5MD50AA
NULL	NULL	1603918	5MD60AA
NULL	NULL	1605509	5AC30HA2