Video-0: Getting Started with SQL

Corresponds to Chapter-0 in The Free SQL Book:
A Tutorial Introduction to SQL
(www.freesqlbook.com)

Preliminary Comments

A. What's an RDBMS?

B. What's SQL?

C. Preview: Sample Queries

D. Hands-On SQL

Pronounce “SQL”

1. Proper pronunciation: 3 letters: **S** – **Q** – **L**
   (Structured Query Language)

2. Ancient history pronunciation: SEQUEL
   (Structured English Query Language)

Why SQL?

"Jobs" ranking - SQL that shines at No. 1.

You're very unlikely to get a job as a pure SQL programmer.

Instead, employers love, love, *love*, seeing SQL skills in tandem with some other language such as Java or C++.

Experienced SQL Users

Many experienced SQL users can skip many/most chapters/videos.

Examine book/video's Table of Contents (on next page).

Notice what topics are covered (and not covered).

Go directly to desired video and/or chapter.

Table of Contents

Target Audience

Super-User (Business User/Expert)

- Accountant
- Financial Analyst
- Engineer
- Data Scientist

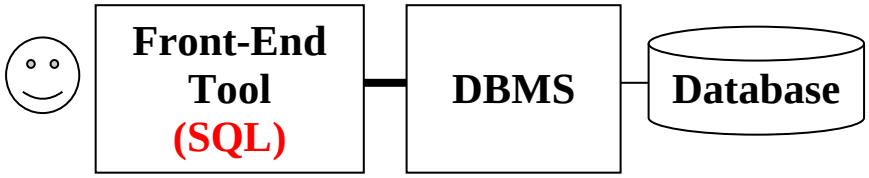
Application Developer (“Techie”)

- Computer Programmer
- System (Database) Analyst/Designer

Rookie (Student) ***

- Future Super-User
- Future Applications Developer

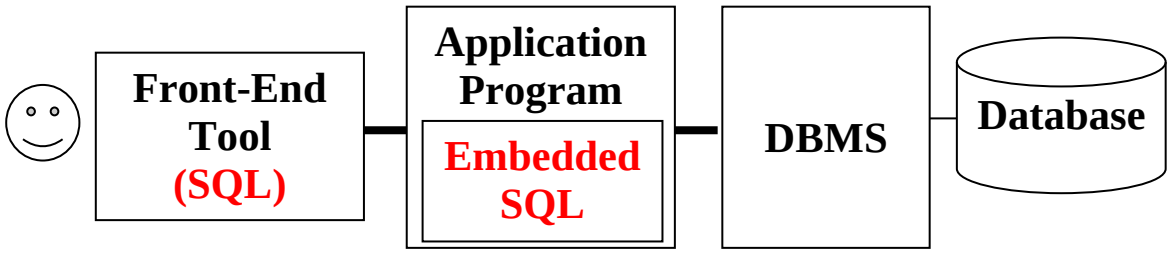
Super-User



Front-End Tool: "Friendly" Query Tool
User points-and-clicks.
Tool generates SQL.

Why learn SQL? Tool has limits, especially for complex queries
User reviews SQL code
User modifies/rewrites SQL code

Application Developer



Front-End Tool: Integrated Development Environment (IDE)
Code Embedded SQL
Database Design
Database Administration

What's (not) in Videos

IN: Videos/book covers "Core" – "Vanilla" – Interactive SQL for **R**DBMS

Data is *Structured*: Rows in a table. (Look like records in a file)
Numbers, Character-Strings, Dates/Time

Also, book has ***optional*** appendices (not presented in videos) that address:
Query Efficiency
Database Analysis Design
Theory behind the Relational Model

- -

NOT IN: Videos/book does **not** cover
How to Embed SQL
How to handle *Semi-Structured and Unstructured Data*

A. What's an RDBMS?

Relational Database Management Systems

Simple Answer: Data is stored in tables.

Ex: Academic Database

Popular Relational Database Systems

- ORACLE
- DB2
- SQL Server
- MYSQL
- SQLite
- PostgreSQL

PRESERVE Table

Sample data from The Nature Conservancy

PRESERVE Table

PNO	PNAME	STATE	ACRES	FEE
5	HASSAYAMPA RIVER	AZ	660	3.00
3	DANCING PRAIRIE	MT	680	0.00
7	MULESHOE RANCH	AZ	49120	0.00
40	SOUTH FORK MADISON	MT	121	0.00
14	MCELWAIN-OLSEN	MA	66	0.00
13	TATKON	MA	40	0.00
9	DAVID H. SMITH	MA	830	0.00
11	MIACOMET MOORS	MA	4	0.00
12	MOUNT PLANTAIN	MA	730	0.00
1	COMERTOWN PRAIRIE	MT	1130	0.00
2	PINE BUTTE SWAMP	MT	15000	0.00
80	RAMSEY CANYON	AZ	380	3.00
10	HOFT FARM	MA	90	0.00
6	PAPAGONIA-SONOITA CREEK	AZ	1200	3.00

“Plug” for the Nature Conservancy

(www.nature.org)

Protects properties from development in most states.

> Nothing wrong with development. We need houses and apartments to live in, maybe a few shopping malls and even a few golf courses.

Some properties are just too beautiful, or too ecologically significant, to be turned into a shopping mall or golf course.

The Nature Conservancy attempts to purchase such endangered properties.
If successful, you and I can take a hike.

B. What's SQL?

SQL is a computer language used within an RDBMS.

Most SQL statements reference a table.

- Create tables (CREATE TABLE statement)
- Insert rows into tables (INSERT statement)
- Update data in tables (UPDATE Statement)
- Delete rows from tables (DELETE Statement)
- Retrieve data from tables (SELECT statement)

Preliminary Observations

- **SQL is easy.**
- **However, Knowing your *data* can be a challenge, and**
- **Knowing your *logic* can be a challenge.**

PRESERVE Table (Details)

PNO	PNAME	STATE	ACRES	FEE
5	HASSAYAMPA RIVER	AZ	660	3.00
3	DANCING PRAIRIE	MT	680	0.00
7	MULESHOE RANCH	AZ	49120	0.00
40	SOUTH FORK MADISON	MT	121	0.00
14	MCELWAIN-OLSEN	MA	66	0.00
13	TATKON	MA	40	0.00
9	DAVID H. SMITH	MA	830	0.00
11	MIACOMET MOORS	MA	4	0.00
12	MOUNT PLANTAIN	MA	730	0.00
1	COMERTOWN PRAIRIE	MT	1130	0.00
2	PINE BUTTE SWAMP	MT	15000	0.00
80	RAMSEY CANYON	AZ	380	3.00
10	HOFT FARM	MA	90	0.00
6	PAPAGONIA-SONOITA CREEK	AZ	1200	3.00

Each **column** has a name: PNO, PNAME, STATE, ACRES, FEE

Eyeball data: Numeric versus character-string data.

“Eyeballing” has limitations: Is PNO numeric or character-string of digits?

Must know **data-type** of each column.

Column Data-Types

Numeric columns
PNO: **INTEGER**
ACRES: **INTEGER**
FEE: **DECIMAL**

Character-string columns
STATE: **CHAR**
PNAME: **VARCHAR**

PNO	PNAME	STATE	ACRES	FEE
5	HASSAYAMPA RIVER	AZ	660	3.00
3	DANCING PRAIRIE	MT	680	0.00
7	MULESHOE RANCH	AZ	49120	0.00
40	SOUTH FORK MADISON	MT	121	0.00
14	MCELWAIN-OLSEN	MA	66	0.00
13	TATKON	MA	40	0.00
9	DAVID H. SMITH	MA	830	0.00
11	MIACOMET MOORS	MA	4	0.00
12	MOUNT PLANTAIN	MA	730	0.00
1	COMERTOWN PRAIRIE	MT	1130	0.00
2	PINE BUTTE SWAMP	MT	15000	0.00
80	RAMSEY CANYON	AZ	380	3.00
10	HOFT FARM	MA	90	0.00
6	PAPAGONIA-SONOITA CREEK	AZ	1200	3.00

What is difference between CHAR versus VARCHAR?

CREATE TABLE Statement

You may (or may not) be able to examine CREATE TABLE statements.

```
CREATE TABLE PRESERVE
(PNO      INTEGER          NOT NULL UNIQUE,
 PNAME    VARCHAR (25)     NOT NULL,
 STATE    CHAR (2)         NOT NULL,
 ACRES    INTEGER          NOT NULL,
 FEE      DECIMAL (5,2)    NOT NULL)
```

CREATE TABLE statement creates an “empty” table.

Sample INSERT statement.

```
INSERT INTO PRESERVE
VALUES (3,   'DANCING PRAIRIE',  'MT',  680,  0.00);
```

Knowing-Your-Data

Know: Table-name + each column's:

- column-name
- data-type
- unique values?
- null values allowed?

How to know your data:

1. Read Documentation [possibly obsolete]

2. Examine CREATE TABLE statement [possibly obsolete]

3. Ask the system (Metadata)

C. Preview - Sample Queries in Chapter 1

"What's going on here?"

Sample Query 1.1: Display all data in the PRESERVE table.

```
SELECT *

FROM PRESERVE
```

PNO	PNAME	STATE	ACRES	FEE
5	HASSAYAMPA RIVER	AZ	660	3.00
3	DANCING PRAIRIE	MT	680	0.00
7	MULESHOE RANCH	AZ	49120	0.00
40	SOUTH FORK MADISON	MT	121	0.00
14	MCELWAIN-OLSEN	MA	66	0.00
13	TATKON	MA	40	0.00
9	DAVID H. SMITH	MA	830	0.00
11	MIACOMET MOORS	MA	4	0.00
12	MOUNT PLANTAIN	MA	730	0.00
1	COMERTOWN PRAIRIE	MT	1130	0.00
2	PINE BUTTE SWAMP	MT	15000	0.00
80	RAMSEY CANYON	AZ	380	3.00
10	HOFT FARM	MA	90	0.00
6	PAPAGONIA-SONOITA CREEK	AZ	1200	3.00

Sample Query 1.2: Display rows where the FEE value equals 3.00. Display all columns in these rows .

```
SELECT *

FROM PRESERVE

WHERE FEE = 3.00
```

PNO	PNAME	STATE	ACRES	FEE
5	HASSAYAMPA RIVER	AZ	660	3.00
80	RAMSEY CANYON	AZ	380	3.00
6	PAPAGONIA-SONOITA CREEK	AZ	1200	3.00

Sample Query 1.3a: Display all information about any nature preserve that is located in Arizona.

```
SELECT *

FROM PRESERVE

WHERE STATE = 'AZ'
```

PNO	PNAME	STATE	ACRES	FEE
5	HASSAYAMPA RIVER	AZ	660	3.00
7	MULESHOE RANCH	AZ	49120	0.00
80	RAMSEY CANYON	AZ	380	3.00
6	PAPAGONIA-SONOITA CREEK	AZ	1200	3.00

Sample Query 1.3b: Display all information about the Ramsey Canyon preserve.

```
SELECT *

FROM PRESERVE

WHERE PNAME = 'RAMSEY CANYON'
```

PNO	PNAME	STATE	ACRES	FEE
80	RAMSEY CANYON	AZ	380	3.00

Note: CHAR and VARCHAR *usually* treated in the same manner.

Sample Query 1.4: For every row in PRESERVE,
display its PNAME, ACRES, and STATE values

```
SELECT PNAME, ACRES, STATE

FROM PRESERVE
```

PNAME	ACRES	STATE
HASSAYAMPA RIVER	660	AZ
DANCING PRAIRIE	680	MT
MULESHOE RANCH	49120	AZ
SOUTH FORK MADISON	121	MT
MCELWAIN-OLSEN	66	MA
TATKON	40	MA
DAVID H. SMITH	830	MA
MIACOMET MOORS	4	MA
MOUNT PLANTAIN	730	MA
COMERTOWN PRAIRIE	1130	MT
PINE BUTTE SWAMP	15000	MT
RAMSEY CANYON	380	AZ
HOFT FARM	90	MA
PAPAGONIA-SONOITA CREEK	1200	AZ

Sample Query 1.5: Display the PNAME and ACRES values for every nature preserve located in Arizona.

```
SELECT PNAME, ACRES

FROM   PRESERVE

WHERE STATE = 'AZ'
```

PNAME	ACRES
HASSAYAMPA RIVER	660
MULESHOE RANCH	49120
RAMSEY CANYON	380
PAPAGONIA-SONOITA CREEK	1200

D. Hands-On SQL: Front-End Tool

SQL-Page

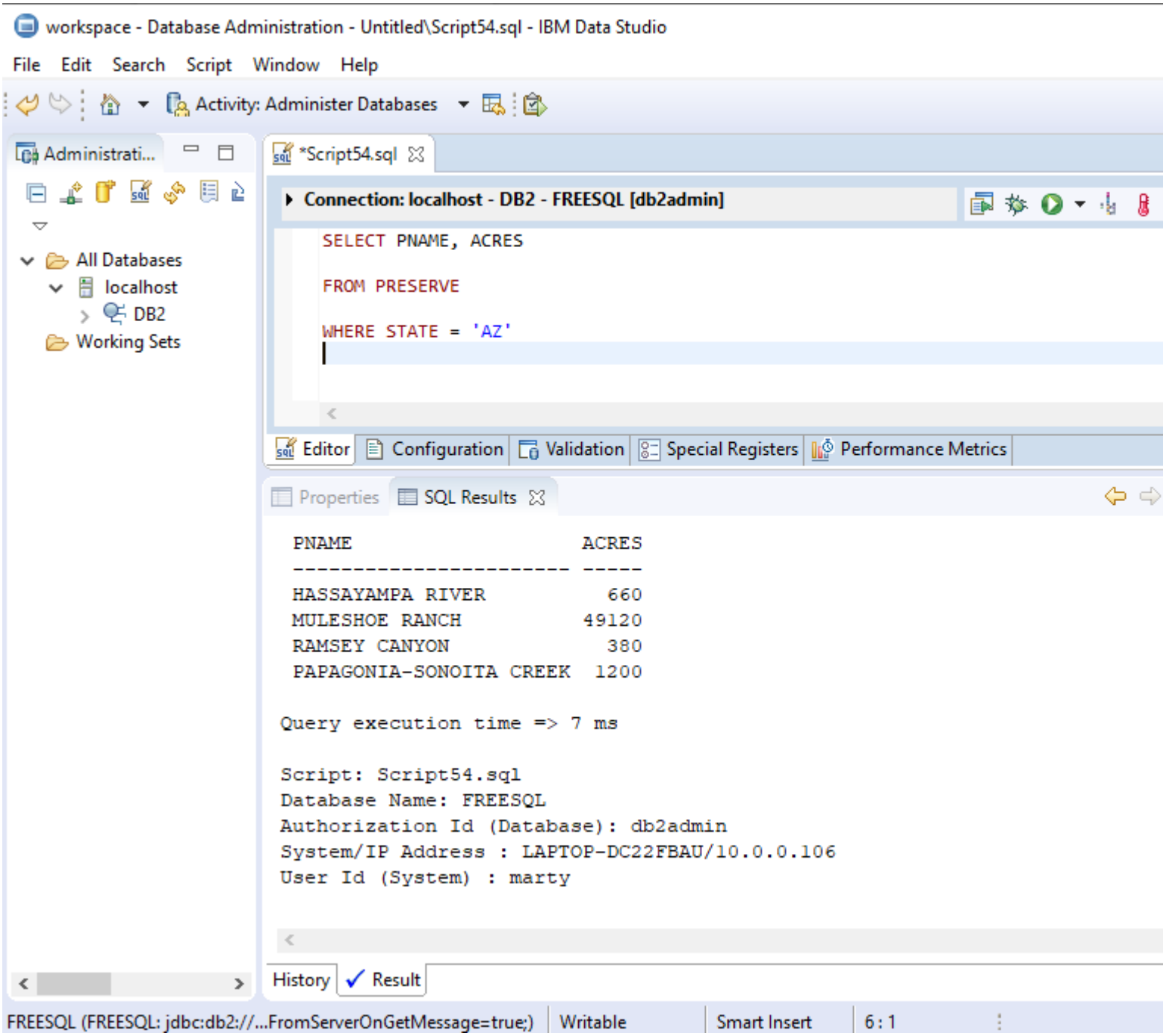
(1) SQL Panel

```
SELECT PNAME, ACRES

FROM PRESERVE

WHERE STATE = 'AZ'
```

(2) Result Panel

PNAME	ACRES
HASSAYAMPA RIVER	660
MULESHOE RANCH	49120
RAMSEY CANYON	380
PAPAGONIA-SONOITA CREEK	1200

SQL-Page: IBM Data Studio



Commercial Software

Figure 0.3a: System Architecture

Figure 0.3b: System Architecture

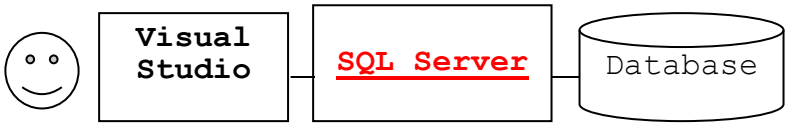


Figure 0.3c: System Architecture

Free SQL Book: "Really" Free

www.freesqlbook.com

No registration – No email request - No cookies, Etc.

After reading this book, if you think it worth at least $5.00,
optionally donate that amount *directly* to:

- Nature Conservancy (www.nature.org)
- World Wildlife Organization (www.worldwildlife.org)
- National Audubon Society (www.audubon.org)

Restrictions

1. Book is protected by a copyright.
2. For-profit training organizations make modest donation. (Details on Pg 2)

This Video (Video-1: SELECT)

Based on Chapter 1 in The FREE SQL Book (www.freesqlbook.com)

The preceding video was based on Chapter 0 in this book.
Rookies are encouraged to watch Video-0 before watching this video.

Topics

- Sample Queries (1.1 – 1.5)
- Nuisance Issues
- Hands-on SQL: The FREESQL Script

Sample Query 1.1: Display all data in the PRESERVE table.

```
SELECT *

FROM PRESERVE
```

PNO	PNAME	STATE	ACRES	FEE
5	HASSAYAMPA RIVER	AZ	660	3.00
3	DANCING PRAIRIE	MT	680	0.00
7	MULESHOE RANCH	AZ	49120	0.00
40	SOUTH FORK MADISON	MT	121	0.00
14	MCELWAIN-OLSEN	MA	66	0.00
13	TATKON	MA	40	0.00
9	DAVID H. SMITH	MA	830	0.00
11	MIACOMET MOORS	MA	4	0.00
12	MOUNT PLANTAIN	MA	730	0.00
1	COMERTOWN PRAIRIE	MT	1130	0.00
2	PINE BUTTE SWAMP	MT	15000	0.00
80	RAMSEY CANYON	AZ	380	3.00
10	HOFT FARM	MA	90	0.00
6	PAPAGONIA-SONOITA CREEK	AZ	1200	3.00

Logic: No WHERE-clause ➔ retrieve all rows

Important Observation: **No row sequence!**

- *In principle, tables do not have any predefined row sequence*.
- No ORDER BY clause (to be presented in Chapter 2).

Incidental Row Sequence ➔ Rows sorted without ORDER BY clause

Observations

Reserved Words (Keywords): SELECT – FROM – WHERE

Termination character:

```
SELECT *
FROM   PRESERVE;
```

Single-line Coding:

```
SELECT * FROM PRESERVE
```

Formatting the Result: SQL does not directly support formatting. Many front-end tools help.

Sample Query 1.2: Display all information about any nature preserve with an admission fee of $3.00. Display all columns in these rows.

```
SELECT *
FROM PRESERVE
WHERE FEE = 3.00
```

PNO	PNAME	STATE	ACRES	FEE
5	HASSAYAMPA RIVER	AZ	660	3.00
80	RAMSEY CANYON	AZ	380	3.00
6	PAPAGONIA-SONOITA CREEK	AZ	1200	3.00

Compare on: = <> < > <= >=

No Punctuation: Minus sign (-) is allowed (WHERE FEE = -3.00).

Know-Your-Data: Mathematical comparison.

WHERE FEE = 3	(valid)
WHERE FEE = 3.0	(valid)
WHERE FEE = 3.000	(valid)

Buzzword: Restrict → Retrieve all columns from a subset of rows

Sample Query 1.3a: Display all information about any nature preserve that is located in Arizona. (I.e., Display just those rows where the STATE value is AZ.)

```
SELECT *
FROM PRESERVE
WHERE STATE = 'AZ'
```

PNO	PNAME	STATE	ACRES	FEE
5	HASSAYAMPA RIVER	AZ	660	3.00
7	MULESHOE RANCH	AZ	49120	0.00
80	RAMSEY CANYON	AZ	380	3.00
6	PAPAGONIA-SONOITA CREEK	AZ	1200	3.00

Syntax: Character-string value enclosed within apostrophes.

Logic: Character-by-character comparison. (Unlike math compare)

Sample Query 1.3b: Display all information about the Ramsey Canyon nature preserve.

```
SELECT *
FROM PRESERVE
WHERE PNAME = 'RAMSEY CANYON'
```

PNO	PNAME	STATE	ACRES	FEE
80	RAMSEY CANYON	AZ	380	3.00

CHAR versus VARCHAR: No difference *for this comparison*

Embedded Blanks: PNAME = 'RAMSEYCANYON' → No Hit
PNAME = 'RAMSEY CANYON' → No Hit

Leading Blanks: PNAME = ' RAMSEY CANYON' → No Hit

Sample Query 1.4: Display its PNAME, ACRES, and STATE values (in that left-to-right column sequence) for all rows in PRESERVE.

```
SELECT PNAME, ACRES, STATE
FROM PRESERVE
```

PNAME	ACRES	STATE
HASSAYAMPA RIVER	660	AZ
DANCING PRAIRIE	680	MT
MULESHOE RANCH	49120	AZ
OUTH FORK MADISON	121	MT
CELWAIN-OLSEN	66	MA
TATKON	40	MA
DAVID H. SMITH	830	MA
MIACOMET MOORS	4	MA
MOUNT PLANTAIN	730	MA
COMERTOWN PRAIRIE	1130	MT
PINE BUTTE SWAMP	15000	MT
RAMSEY CANYON	380	AZ
HOFT FARM	90	MA
PAPAGONIA-SONOITA CREEK	1200	AZ

Buzzword - Project: Retrieve a subset of columns from all rows.

Sample Query 1.5: Display the PNAME and ACRES values of every nature preserve that is located in Arizona.

```
SELECT PNAME, ACRES
FROM  PRESERVE
WHERE STATE = 'AZ'
```

PNAME	ACRES
HASSAYAMPA RIVER	660
MULESHOE RANCH	49120
RAMSEY CANYON	380
PAPAGONIA-SONOITA CREEK	1200

Syntax & Logic: *Nothing new!* "Restrict + Project"

Observation: Result table is a "subset" of PRESERVE table

Internal Storage of CHAR versus VARCHAR

Figure 1.1a: Outside View of STATENAME Column

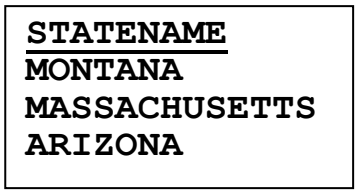


Figure 1.1b: Inside View stored as CHAR (14)

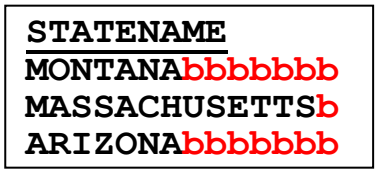
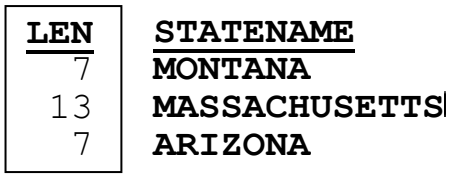


Figure 1.1c: Inside View stored as VARCHAR (14)



Nuisance Issues with Character-Strings

Issus pertain to different behavior on different systems.

Simple problem after you learn your system.

Observations about PNAME and STATE columns in PRESERVE table.

- All alphabetical characters are upper-case.
- No character-string value has leading spaces.

  No value like: ' DANCING PRAIRIE'
- No variable-length character-string value has trailing spaces.

  No value like: 'DANCING PRAIRIE '

Case-Sensitivity

Character-strings can contain upper-case, lower-case, or mixed-case values.
STATE column has upper-case values.

Sample Query 1.3a specified WHERE-clause with upper-case letters.

```
WHERE STATE = 'AZ'
```

Careless Coding: What happens if WHERE-clause looks like:

```
WHERE STATE = 'az'
WHERE STATE = 'Az'
WHERE STATE = 'aZ'
```

Answer for ORACLE, DB2, SQL Server: "No-hit" (no rows returned)

Answer for MYSQL: "Hit" – Four rows returned.

Chapter 2 will discuss collating sequence for character-string values.

Chapter 6 will discuss UPPER and LOWER functions.

Leading Blanks

Leading blanks should (almost) never be stored in any character-string column.

During data input, a typo may specify leading blanks.

However, DBA usually takes some action that prevents leading blanks.

Careless Coding: What happens if WHERE-clause looks like:

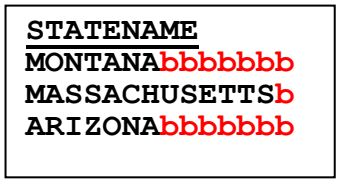
WHERE PNAME = ' DANCING PRAIRIE'

Result: No hit on all systems

Chapter 6 will discuss LTRIM function.

Trailing Blanks: CHAR columns

CHAR columns frequently have trailing blanks.



All systems will select the MONTANA row on:

WHERE STATENAME = '**MONTANA**'

Systems have some method to deal with trailing blanks.
Not all systems use the same method.
System specific methods for CHAR comparison will be described later.

Careless Coding: What happens if WHERE-clause looks like:

WHERE STATENAME = '**MONTANA** '

Answer for ORACLE, DB2, SQL Server: Montana row is returned.

Answer for MYSQL: Montana row is not returned.

Trailing Blanks: VARCHAR columns

VARCHAR columns rarely have trailing blanks (as shown below).

LEN	VSTATENAME
7	MONTANA
13	MASSACHUSETTS
7	ARIZONA

All systems will select the MONTANA row on:

WHERE STATENAME = '**MONTANA**'

Systems have some method to deal with trailing blanks.
Not all systems use the same method.
System specific methods for VARCHAR comparison will be described later.

Careless Coding: What happens if WHERE-clause looks like:

WHERE STATENAME = '**MONTANA** '

Answer for DB2 and SQL Server: Montana row is returned.

Answer for ORACLE and MYSQL: Montana row is not returned.

Criteria

Your SQL statements should be:

- **Correct *****
- Efficient: Makes no sense to do the wrong thing fast!
- Friendly: Another user may modify your SQL statement

Hands-on SQL

Create FreeSQL Sample Tables (Book-Append-I)

Access to Front-End Tool + RDBMS (Book-Append-II)

SQL Scripts

SQL Script is a collection of SQL statements that are:
separated by semicolons (;) and
executed as a single unit.

Example:

```
SELECT * FROM PRESERVE
WHERE STATE =   'AZ';

SELECT * FROM PRESERVE
WHERE STATE =   'MA';

SELECT * FROM PRESERVE
WHERE STATE =   'MT';
```

SQL Scripts may contain a variety of SQL statements, such as:

CREATE TABLE
INSERT
COMMIT
DROP TABLE

Create FreeSQL Sample Tables

CHPT-1-5-Script

```
-- This script creates the PRESERVE and EMPLOYEE tables.
-- Only tables referenced in Chapters 1-5.
-- SQL Server users remove COMMIT statements from this script.

DROP TABLE PRESERVE;
DROP TABLE EMPLOYEE;

CREATE TABLE PRESERVE
(PNO        INTEGER          NOT NULL UNIQUE,
 PNAME      VARCHAR (25)     NOT NULL,
 STATE      CHAR (2)         NOT NULL,
 ACRES      INTEGER          NOT NULL,
 FEE        DECIMAL (5,2)    NOT NULL);


INSERT INTO PRESERVE
      VALUES (5,  'HASSAYAMPA RIVER',  'AZ', 660,      3.00);
INSERT INTO PRESERVE
      VALUES (3,  'DANCING PRAIRIE',      'MT', 680,      0.00);
INSERT INTO PRESERVE
      VALUES (7,  'MULESHOE RANCH',    'AZ', 49120,    0.00);
-- Eleven more INSERT statements

COMMIT;   -- SQL Server users should remove this statement

CREATE TABLE EMPLOYEE
 -- . . . . . Etc……..
```

This Video (Video-2: ORDER BY)

Based on Chapter 2 in The FREE SQL Book. (www.freesqlbook.com)

Objective: Result table with rows in desired row sequence.
(Cannot store base tables in row sequence,)

Sample Queries (2.1 – 2.9)

Related Topics:

- "Sequence" versus "Sort"
- Incidental Sort
- Collating Sequence
- Deterministic vs Non-Deterministic Statements
- "First" N Rows

Order by a Single Column

Sample Query 2.1: Retrieve the STATE, PNO, and PNAME values from all rows in the PRESERVE table. Display rows sorted by STATE values in ascending sequence.

```
SELECT STATE, PNO, PNAME
FROM   PRESERVE
ORDER BY STATE
```

STATE	PNO	PNAME
AZ	5	HASSAYAMPA RIVER
AZ	7	MULESHOE RANCH
AZ	80	RAMSEY CANYON
AZ	6	PAPAGONIA-SONOITA CREEK
MA	14	MCELWAIN-OLSEN
MA	13	TATKON
MA	9	DAVID H. SMITH
MA	11	MIACOMET MOORS
MA	12	MOUNT PLANTAIN
MA	10	HOFT FARM
MT	3	DANCING PRAIRIE
MT	40	SOUTH FORK MADISON
MT	1	COMERTOWN PRAIRIE
MT	2	PINE BUTTE SWAMP

Ascending is default sequence. ORDER BY STATE ASC

Know-Your-Data: STATE is not a unique column.
No assumptions about a second-level sort sequence.

Order by Multiple Columns

Sample Query 2.2: Display STATE, PNO, and PNAME values for every nature preserve. Sort the result by PNO within STATE. (I.e., STATE is the primary sort column, and PNO is the secondary sort column.)

```
SELECT STATE, PNO, PNAME
FROM   PRESERVE
ORDER BY STATE, PNO
```

STATE	PNO	PNAME
AZ	5	HASSAYAMPA RIVER
AZ	6	PAPAGONIA-SONOITA CREEK
AZ	7	MULESHOE RANCH
AZ	80	RAMSEY CANYON
MA	9	DAVID H. SMITH
MA	10	HOFT FARM
MA	11	MIACOMET MOORS
MA	12	MOUNT PLANTAIN
MA	13	TATKON
MA	14	MCELWAIN-OLSEN
MT	1	COMERTOWN PRAIRIE
MT	2	PINE BUTTE SWAMP
MT	3	DANCING PRAIRIE
MT	40	SOUTH FORK MADISON

Know-Your-Data: PNO is a unique column.
Hence, no duplicates for (STATE, PNO)

Descending Sort: DESC

Sample Query 2.3: Display the PNO, PNAME, and ACRES values acres for all nature preserves that are located in Arizona. Display the result by PNO values in descending sequence.

```
SELECT PNO, PNAME, ACRES
FROM   PRESERVE
WHERE STATE = 'AZ'
ORDER BY PNO DESC
```

PNO	PNAME	ACRES
80	RAMSEY CANYON	380
7	MULESHOE RANCH	49120
6	PAPAGONIA-SONOITA CREEK	1200
5	HASSAYAMPA RIVER	660

ASC or DESC can be specified for multiple columns.

ORDER BY COL1 ASC, COL2 DESC, COL3 ASC, COL4 DESC

Order by Column-Number

Sample Query 2.4: Display the PNO, ACRES. and PNAME values of all nature preserves located in Arizona. Sort the result by the third column.

```
SELECT PNO, ACRES, PNAME
FROM   PRESERVE
WHERE STATE = 'AZ'
ORDER BY 3
```

```
PNO   ACRES      PNAME
  5     660      HASSAYAMPA RIVER
  7   49120      MULESHOE RANCH
  6    1200      PAPAGONIA-SONOITA CREEK
 80     380      RAMSEY CANYON
```

Specifying a column-number is acceptable for a one-time ad hoc query.

Order by a Non-Displayed Column

Sample Query 2.5: Display the PNO and PNAME values for all preserves. Display the result by ACRES in descending sequence.

```
SELECT PNO, PNAME
FROM   PRESERVE
ORDER BY ACRES DESC
```

PNO	PNAME
7	MULESHOE RANCH
2	PINE BUTTE SWAMP
6	PAPAGONIA-SONOITA CREEK
1	COMERTOWN PRAIRIE
9	DAVID H. SMITH
12	MOUNT PLANTAIN
3	DANCING PRAIRIE
5	HASSAYAMPA RIVER
80	RAMSEY CANYON
40	SOUTH FORK MADISON
10	HOFT FARM
14	MCELWAIN-OLSEN
13	TATKON
11	MIACOMET MOORS

Better Example: Sort by SALARY; but do not display SALARY.

```
SELECT ENAME
FROM   EMPLOYEE
ORDER BY SALARY
```

Tutorial Example

Sample Query 2.6: Display the STATE, FEE, and PNAME values from all rows in the PRESERVE table where:

- STATE is the 1st level sort column (ascending)
- FEE is the 2nd level sort column (descending)
- PNAME is the 3rd level sort column (descending)

```
SELECT STATE, FEE, PNAME
FROM   PRESERVE
ORDER BY STATE ASC,  FEE DESC,  3 DESC
```

STATE	FEE	PNAME
AZ	3.00	RAMSEY CANYON
AZ	3.00	PAPAGONIA-SONOITA CREEK
AZ	3.00	HASSAYAMPA RIVER
AZ	0.00	MULESHOE RANCH
MA	0.00	TATKON
MA	0.00	MOUNT PLANTAIN
MA	0.00	MIACOMET MOORS
MA	0.00	MCELWAIN-OLSEN
MA	0.00	HOFT FARM
MA	0.00	DAVID H. SMITH
MT	0.00	SOUTH FORK MADISON
MT	0.00	PINE BUTTE SWAMP
MT	0.00	DANCING PRAIRIE
MT	0.00	COMERTOWN PRAIRIE

“Sequence” versus “Sort”

Query objective articulated as:

> **“Sort rows in by _____”**
>
> Sorting is an internal (under-the-hood) process that may or may not be used to satisfy some sequence objective.

> **“Display rows in ascending sequence by _____”**
>
> Mrate articulation of the query objective.
> It states “what” to do, not “how to” do it.

Future query objectives will continue to state: “*Sort* the rows by ___”

Incidental Sort

Sample Query 2.7: Display the PNO values for all preserves.
[Observe: No request for row sequenced.]

```
SELECT PNO
FROM   PRESERVE
```

PNO
1
2
3
5
6
7
9
10
11
12
13
14
40
80

Important: No ORDER BY clause → Cannot predict row sequence.

Incidental sort may or may not happen in the future.

ASCII Collating Sequence

Blank character
Most (but not all) special characters
Digits
Uppercase letters
Lowercase letters
A few special characters

Unsorted
Zeek
jessie
JULIE
77aaaaaaaAAAA
JEssie
julie
Jessie
3M
!!!FIDO!!!
JULIe

Sorted (ASCII)
!!!FIDO!!!
3M
77aaaaaaaAAAA
JEssie
JULIE
JULIe
Jessie
Zeek
jessie
julie

DB2 mainframe uses EBCDIC sequence

Character-String Comparison

Sample Query 2.8: Display all information about any nature preserve with a PNAME value that follows the letter R in alphabetical sequence.

```
SELECT *
FROM   PRESERVE
WHERE PNAME > 'R'
```

PNO	PNAME	STATE	ACRES	FEE
40	SOUTH FORK MADISON	MT	121	0.00
13	TATKON	MA	40	0.00
80	RAMSEY CANYON	AZ	380	3.00

Logic. Must understand collating sequence

RAMSEY CANYON has 13 characters.
System pads 'R' with 12 trailing blanks.

System compares: 'R '
'RAMSEY CANYON'

RAMSEY CANYON appears in the result because its second character (A) sorts after the blank character.

Deterministic versus Non-Deterministic Statements

Example-1: SELECT STATE, PNO, PNAME
FROM PRESERVE
ORDER BY STATE

Multiple correct result tables → Non-deterministic statement

Observe: Two correct results show below.

STATE	PNO	PNAME
AZ	5	HASSAYAMPA RIVER
AZ	7	MULESHOE RANCH
AZ	80	RAMSEY CANYON
AZ	6	PAPAGONIA-SONOITA CREEK

. . . . . .

STATE	PNO	PNAME
AZ	80	RAMSEY CANYON
AZ	5	HASSAYAMPA RIVER
AZ	7	MULESHOE RANCH
AZ	6	PAPAGONIA-SONOITA CREEK

. . . . . .

Non-deterministic is not necessarily bad.

Chapter 10.5: DATE Functions are non-deterministic.

Return different results on different days.

.

Example-2: Deterministic statement.

```
SELECT STATE, PNO, PNAME
FROM   PRESERVE
ORDER BY STATE, PNO
```

```
STATE     PNO    PNAME
   AZ       5    HASSAYAMPA RIVER
   AZ       6    PAPAGONIA-SONOITA CREEK
   AZ       7    MULESHOE RANCH
   AZ      80    RAMSEY CANYON
. . . . . . . . . . . . . . . . . . . .
```

Example-3: Non-deterministic statement

```
SELECT STATE, PNAME
FROM   PRESERVE
ORDER BY STATE, PNAME
```

```
STATE     PNAME
   AZ     HASSAYAMPA RIVER
   AZ     MULESHOE RANCH
   AZ     PAPAGONIA-SONOITA CREEK
   AZ     RAMSEY CANYON
   . . . . . . . . . . . . . . . . .
```

Deterministic result by including the PNO in the ORDER BY clause.

```
SELECT STATE, PNAME
FROM PRESERVE
ORDER BY STATE, PNAME, PNO
```

Note: PNO is not displayed.

”First” N Rows (RDBMS Specific)

Sample Query 2.9: Display PNO and ACRES from “first” three rows. Sort the result by PNO values. The result table should look like:

PNO	ACRES
1	1130
3	680
9	830

DB2 and MYSQL

```
SELECT  PNO,  ACRES

FROM PRESERVE
ORDER BY PNO
```

DB2 and ORACLE

```
SELECT PNO, ACRES
FROM PRESERVE
ORDER BY PNO
FETCH FIRST 3 ROWS ONLY
```

SQL Server

```
SELECT TOP (3) PNO, ACRES
FROM PRESERVE
ORDER BY PNO
```

This Video (Video-3: DISTINCT)

Based on Chapter 3 in The FREE SQL Book. (www.freesqlbook.com)

Objective: Remove duplicate rows from result table.

Sample Queries (3.1 – 3.5)

Other Topics:

- Qualified Table Names
- Metadata

Duplicate Rows?

Base Tables: Strong Design Guideline: Each row should be distinct.

All FREESQL tables contain distinct rows.

Result Tables: Duplicate rows can appear → can be confusing.

DISTINCT removes duplicate rows.

Duplicate Rows in a Single-Column Result Table

Know-Your-Data: STATE may contain duplicate values.

Sample Query 3.1: Display the STATE column in every row of the PRESERVE table.

```
SELECT STATE
FROM   PRESERVE
```

STATE
AZ
MT
AZ
MT
MA
MA
MA
MA
MA
MT
MT
AZ
MA
AZ

DISTINCT

Sample Query 3.2: Display the STATE column in every row of the PRESERVE table. Do not display duplicate rows.

```
SELECT DISTINCT STATE
FROM   PRESERVE
```

STATE
AZ
MA
MT

Observation: Incidental sort - No ORDER BY clause

Duplicate Rows in a Multi-Column Result Table

Know-Your-Data: Pairs of (STATE, FEE) may contain duplicate values.

Sample Query 3.3: Display the STATE and FEE values for every row in the PRESERVE table.

```
SELECT STATE, FEE
FROM   PRESERVE
ORDER BY STATE, FEE
```

STATE	FEE
AZ	0.00
AZ	3.00
AZ	3.00
AZ	3.00
MA	0.00
MA	0.00
MA	0.00
MA	0.00
MA	0.00
MA	0.00
MT	0.00
MT	0.00
MT	0.00
MT	0.00

,

DISTINCT

Sample Query 3.4: Display distinct pairs of STATE and FEE values from PRESERVE. Sort the result by FEE within STATE.

```
SELECT DISTINCT STATE, FEE
FROM   PRESERVE
ORDER BY STATE, FEE
```

STATE	FEE
AZ	0.00
AZ	3.00
MA	0.00
MT	0.00

Specify DISTINCT once. Want distinct rows (not columns)

Error: SELECT DISTINCT COL1, COL2, DISTINCT COL3
FROM _____

Know-Your-Data

No need to specify DISTINCT if:

- The SELECT-clause references a unique column (e.g., PNO).
- The result table contains just one row (WHERE PNO = 40)

Sample Query 3.5: Display PNO and ACRES values of nature preserves located in Montana. [Do not display duplicate rows.]

```
SELECT PNO, ACRES
FROM   PRESERVE
WHERE STATE = 'MT'
```

PNO	ACRES
3	680
40	121
1	1130
2	15000

Selecting PNO →No need to specify DISTINCT.

Question: S*hould* we always specify DISTINCT?

Qualified (Two-Part) Table-Names

Assume you are Jacqueline Juniper.
Your user-id is JJ011019.
You create a table called PRESERVE and insert some rows into it.

You (Jacqueline) execute:
```
SELECT *
FROM PRESERVE
```

System executes:
```
SELECT *
FROM JJ011019.PRESERVE
```

Other users: Josephine Violet (user-id is JV061317)
Johnny Trouble (user-id is JT051015)

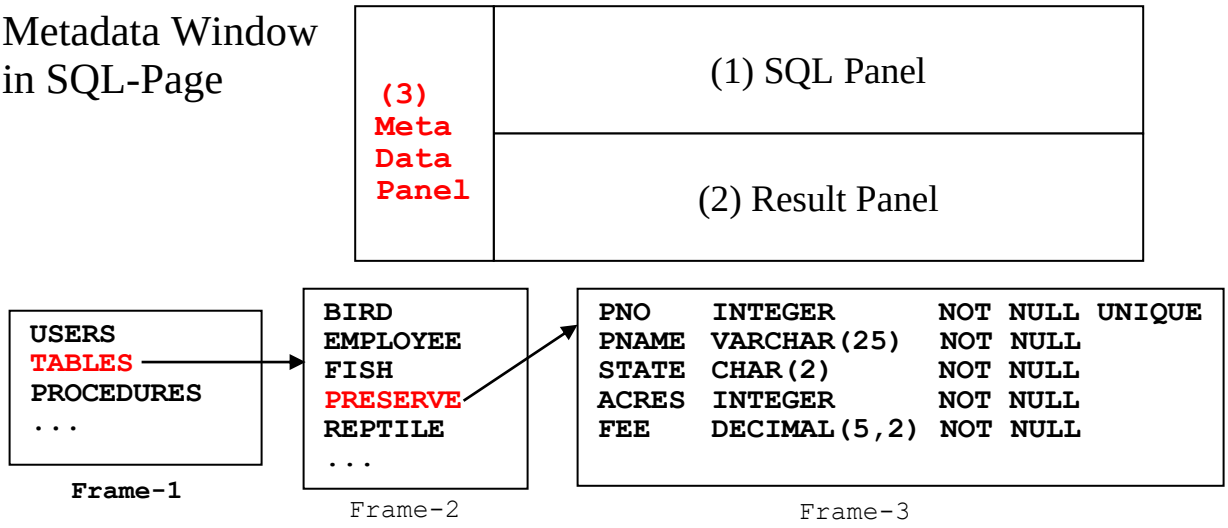
Josephine and Johnny also create tables called PRESERVE.
JV061317.PRESERVE
JT051015.PRESERVE

With permission, you (Jacqueline) access Josephine's PRESERVE table.

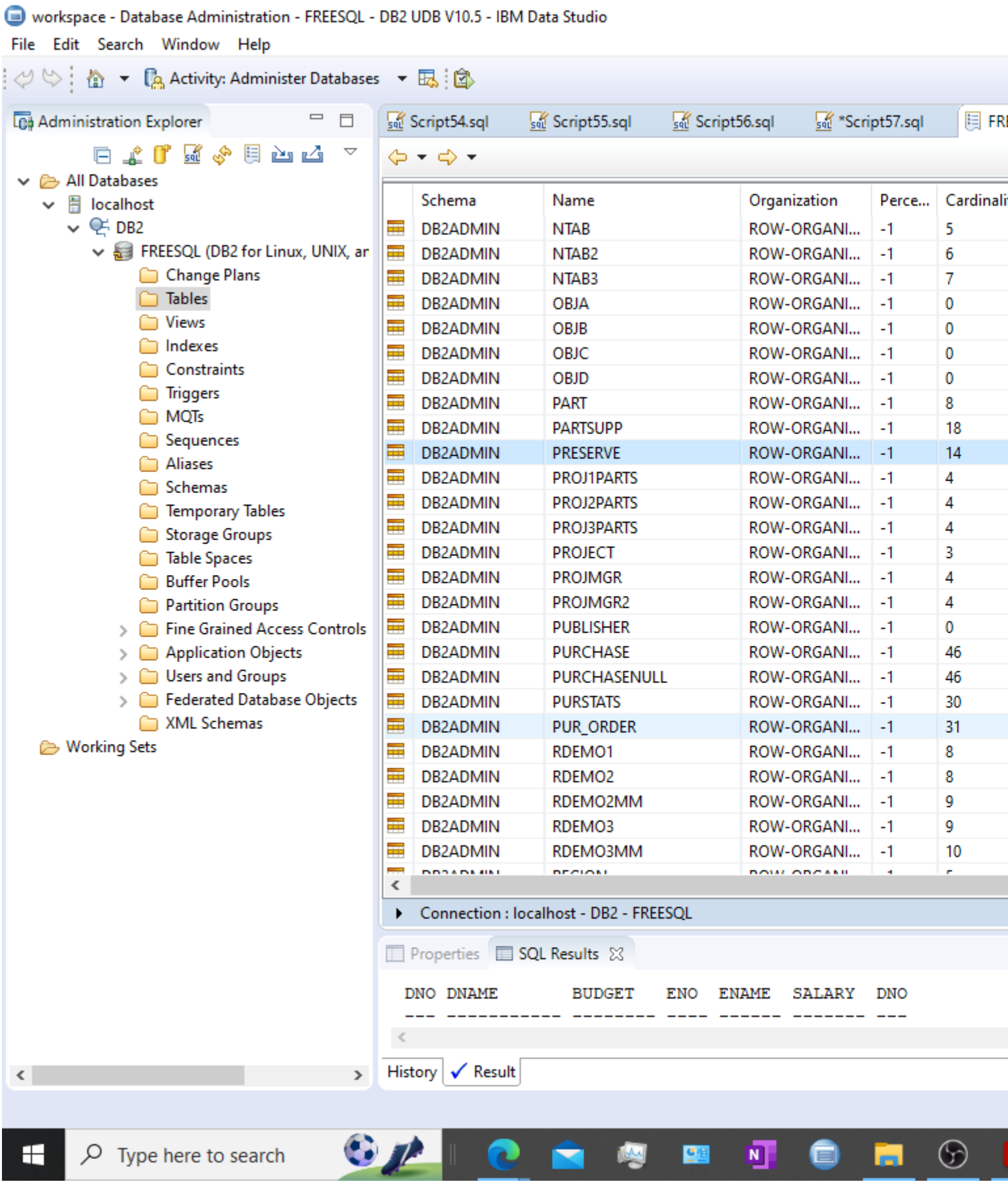
```
SELECT *
FROM JV061317.PRESERVE
```

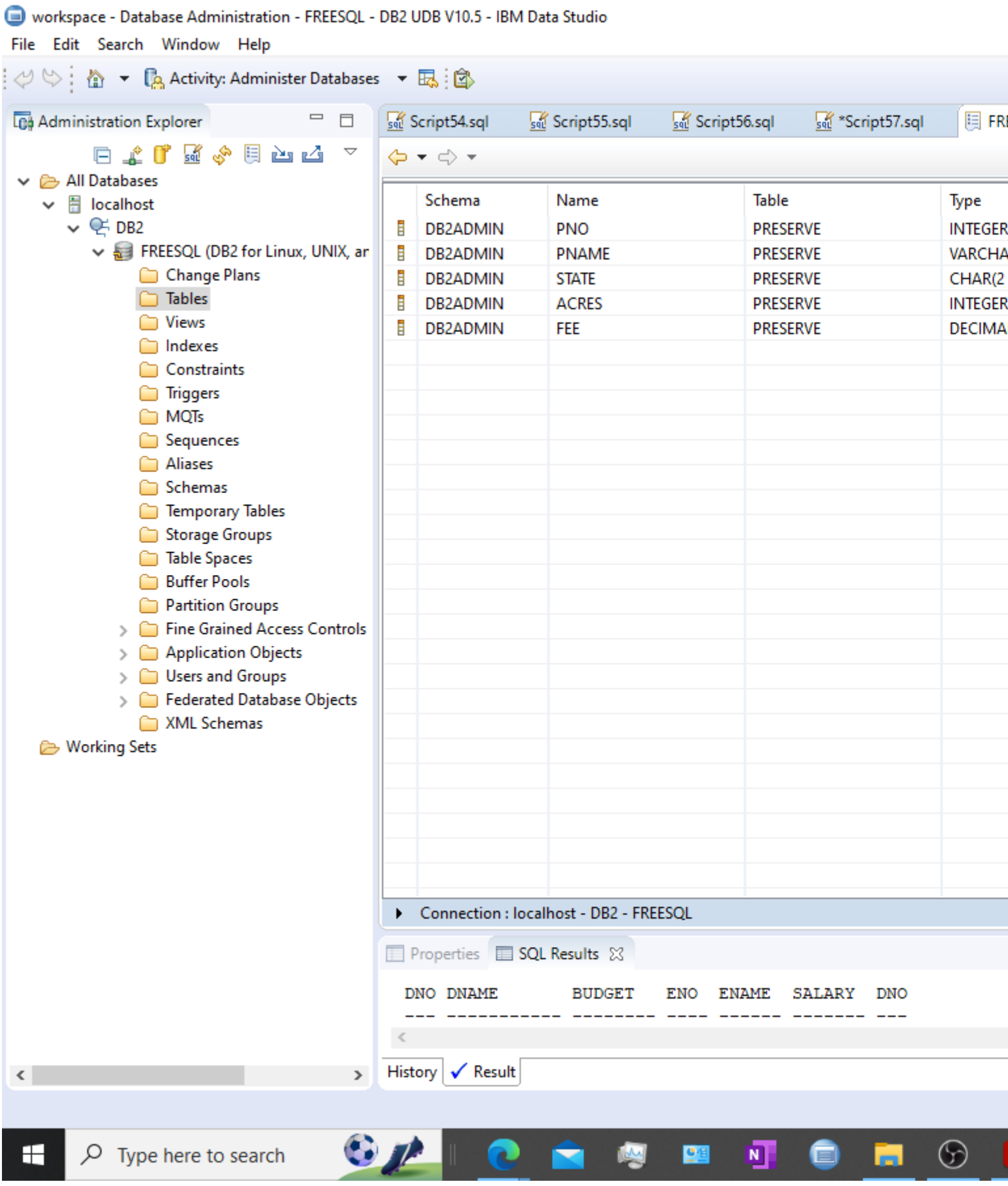
Metadata: Data about Data

Metadata Window in SQL-Page



Metadata is stored in system's *Data Dictionary*.


workspace - Database Administration - FREESQL - DB2 UDB V10.5 - IBM Data Studio
File Edit Search Window Help
Activity: Administer Databases
Administration Explorer
All Databases
localhost
DB2
FREESQL (DB2 for Linux, UNIX, ar
Change Plans
Tables
Views
Indexes
Constraints
Triggers
MQTs
Sequences
Aliases
Schemas
Temporary Tables
Storage Groups
Table Spaces
Buffer Pools
Partition Groups
Fine Grained Access Controls
Application Objects
Users and Groups
Federated Database Objects
XML Schemas
Working Sets
Script54.sql
Script55.sql
Script56.sql
*Script57.sql
Schema Name Organization Perce... Cardinali
DB2ADMIN NTAB ROW-ORGANI... -1 5
DB2ADMIN NTAB2 ROW-ORGANI... -1 6
DB2ADMIN NTAB3 ROW-ORGANI... -1 7
DB2ADMIN OBJA ROW-ORGANI... -1 0
DB2ADMIN OBJB ROW-ORGANI... -1 0
DB2ADMIN OBJC ROW-ORGANI... -1 0
DB2ADMIN OBJD ROW-ORGANI... -1 0
DB2ADMIN PART ROW-ORGANI... -1 8
DB2ADMIN PARTSUPP ROW-ORGANI... -1 18
DB2ADMIN PRESERVE ROW-ORGANI... -1 14
DB2ADMIN PROJ1PARTS ROW-ORGANI... -1 4
DB2ADMIN PROJ2PARTS ROW-ORGANI... -1 4
DB2ADMIN PROJ3PARTS ROW-ORGANI... -1 4
DB2ADMIN PROJECT ROW-ORGANI... -1 3
DB2ADMIN PROJMGR ROW-ORGANI... -1 4
DB2ADMIN PROJMGR2 ROW-ORGANI... -1 4
DB2ADMIN PUBLISHER ROW-ORGANI... -1 0
DB2ADMIN PURCHASE ROW-ORGANI... -1 46
DB2ADMIN PURCHASENULL ROW-ORGANI... -1 46
DB2ADMIN PURSTATS ROW-ORGANI... -1 30
DB2ADMIN PUR_ORDER ROW-ORGANI... -1 31
DB2ADMIN RDEMO1 ROW-ORGANI... -1 8
DB2ADMIN RDEMO2 ROW-ORGANI... -1 8
DB2ADMIN RDEMO2MM ROW-ORGANI... -1 9
DB2ADMIN RDEMO3 ROW-ORGANI... -1 9
DB2ADMIN RDEMO3MM ROW-ORGANI... -1 10
Connection : localhost - DB2 - FREESQL
Properties
SQL Results
DNO DNAME BUDGET ENO ENAME SALARY DNO
--- ---------- -------- ---- ------ -------- ---
History
Result
Type here to search


workspace - Database Administration - FREESQL - DB2 UDB V10.5 - IBM Data Studio
File Edit Search Window Help
Activity: Administer Databases
Administration Explorer
All Databases
localhost
DB2
FREESQL (DB2 for Linux, UNIX, ar
Change Plans
Tables
Views
Indexes
Constraints
Triggers
MQTs
Sequences
Aliases
Schemas
Temporary Tables
Storage Groups
Table Spaces
Buffer Pools
Partition Groups
Fine Grained Access Controls
Application Objects
Users and Groups
Federated Database Objects
XML Schemas
Working Sets
Script54.sql
Script55.sql
Script56.sql
*Script57.sql
Schema Name Table Type
DB2ADMIN PNO PRESERVE INTEGER
DB2ADMIN PNAME PRESERVE VARCHA
DB2ADMIN STATE PRESERVE CHAR(2
DB2ADMIN ACRES PRESERVE INTEGER
DB2ADMIN FEE PRESERVE DECIMA
Connection : localhost - DB2 - FREESQL
Properties
SQL Results
DNO DNAME BUDGET ENO ENAME SALARY DNO
--- ---------- -------- ---- ------ ------- ---
History
Result
Type here to search

Concluding Observation: Uniqueness

We know that PNO is a unique column.

Some tables have multiple unique columns. Is PNAME is unique?

A combination of non-unique columns may be unique.

Assume STATE is non-unique, and
PNAME is non-unique.
Is combination (STATE, PNAME) unique?

This Video (Video-4: AND-OR-NOT)

Based on Chapter 4 in The FREE SQL Book. (www.freesqlbook.com)

Objective: Introduce Boolean Logic

Examples: WHERE STATE = 'AZ' AND FEE = 0.00

WHERE STATE = 'AZ' OR FEE = 0.00

WHERE NOT STATE = 'AZ'

Topics: Section-A: Fundamental Boolean Connectors

Section-B: Mixing Different Boolean Connectors

Section-C: Logically Equivalent WHERE-Clauses

A. Fundamental Boolean Connectors: AND

Sample Query 4.1: Display all information about any nature preserve that is located in Arizona and has no admission fee.

```
SELECT *
FROM   PRESERVE
WHERE STATE = 'AZ' AND FEE = 0.00
```

PNO	PNAME	STATE	ACRES	FEE
7	MULESHOE RANCH	AZ	49120	0.00

Truth Table for AND:

C1	C2	C1 AND C2
T	T	T
T	F	F
F	T	F
F	F	F

Sample Query 4.2: Display the PNO, PNAME, and ACRES values for any nature preserve with an ACRES value that is strictly between 90 and 1200.

```
SELECT *
FROM   PRESERVE
WHERE  ACRES > 90 AND ACRES < 1200
```

PNO	PNAME	ACRES
5	HASSAYAMPA RIVER	660
3	DANCING PRAIRIE	680
40	SOUTH FORK MADISON	121
9	DAVID H. SMITH	830
12	MOUNT PLANTAIN	730
1	COMERTOWN PRAIRIE	1130
80	RAMSEY CANYON	380

Sample Query 4.3: Display the PNO, PNAME, FEE and ACRES values of all nature preserves that are located in Arizona, have a non-zero admission fee, and are greater than or equal to 660 acres, and less than or equal to 1200 acres.

```
SELECT  PNO, PNAME, FEE, ACRES
FROM    PRESERVE
WHERE   STATE = 'AZ'
AND     FEE <> 0.00
AND     ACRES >= 660
AND     ACRES <= 1200
```

PNO	PNAME	FEE	ACRES
5	HASSAYAMPA RIVER	3.00	660
6	PAPAGONIA-SONOITA CREEK	3.00	1200

OR-Connector

Sample Query 4.4: Display the PNO, PNAME, and STATE values of all nature preserves that are located in Arizona or Montana.

```
SELECT PNO, PNAME, STATE
FROM   PRESERVE
WHERE  STATE = 'AZ'
OR     STATE = 'MT'
```

PNO	PNAME	STATE
5	HASSAYAMPA RIVER	AZ
3	DANCING PRAIRIE	MT
7	MULESHOE RANCH	AZ
40	SOUTH FORK MADISON	MT
1	COMERTOWN PRAIRIE	MT
2	PINE BUTTE SWAMP	MT
80	RAMSEY CANYON	AZ
6	PAPAGONIA-SONOITA CREEK	AZ

Truth Table for OR:

C1	C2	C1 OR C2
T	T	T
T	F	T
F	T	T
F	F	F

OR Means "Inclusive OR"

Sample Query 4.5: Display the PNAME, ACRES, and STATE value of any preserve that is located in Arizona or has more than 1000 acres.

```
SELECT PNAME, ACRES, STATE
FROM   PRESERVE
WHERE  STATE = 'AZ'
OR     ACRES > 1000
```

PNAME	ACRES	STATE	
HASSAYAMPA RIVER	660	AZ	
MULESHOE RANCH	49120	AZ	←
COMERTOWN PRAIRIE	1130	MT	
PINE BUTTE SWAMP	15000	MT	
RAMSEY CANYON	380	AZ	
PAPAGONIA-SONOITA CREEK	1200	AZ	←

Inclusive-OR: Select rows matching
One, or the other, or both conditions

Sample Query 4.6: Display the PNO and PNAME values for any nature preserve that has a PNO value equal to any of the values {3, 4, 7, 12, 40}

```
SELECT   PNO, PNAME
FROM     PRESERVE
WHERE    PNO = 3
OR       PNO = 4
OR       PNO = 7
OR       PNO = 12
OR       PNO = 40
```

PNO	PNAME
3	DANCING PRAIRIE
7	MULESHOE RANCH
40	SOUTH FORK MADISON
12	MOUNT PLANTAIN

NOT

Sample Query 4.7: Display the PNAME and STATE values of all nature preserves that are not located in Massachusetts.

```
SELECT   PNAME, STATE
FROM     PRESERVE
WHERE    NOT STATE = 'MA'
```

PNAME	STATE
HASSAYAMPA RIVER	AZ
DANCING PRAIRIE	MT
MULESHOE RANCH	AZ
SOUTH FORK MADISON	MT
COMERTOWN PRAIRIE	MT
PINE BUTTE SWAMP	MT
RAMSEY CANYON	AZ
PAPAGONIA-SONOITA CREEK	AZ

Truth Table for NOT:

C1	NOT C1
T	F
F	T

B. Mixing Different Boolean Connectors

Parenthesis dictate order of evaluation.

SQ 4.8: ACRES > 1000 **OR** (STATE = 'AZ' **AND** FEE = 3.00)

SQ 4.9: (ACRES > 1000 **OR** STATE = 'AZ') **AND** FEE = 3.00

These WHERE-clauses are not equivalent. They return different results.

- -

What if you do not specify any parentheses?

ACRES > 1000 OR STATE = 'AZ' AND FEE = 3.00

Sample Query 4.8: Display the PNAME, STATE, FEE, and ACRES values of any nature preserve with more than 1000 acres, or any Arizona preserve with a $3.00 admission fee.

```
SELECT PNAME, STATE, FEE, ACRES
FROM   PRESERVE
WHERE  ACRES > 1000 OR (STATE = 'AZ' AND FEE = 3.00)
```

PNAME	STATE	FEE	ACRES
HASSAYAMPA RIVER	AZ	3.00	660
MULESHOE RANCH	AZ	0.00	49120
COMERTOWN PRAIRIE	MT	0.00	1130
PINE BUTTE SWAMP	MT	0.00	15000
RAMSEY CANYON	AZ	3.00	380
PAPAGONIA-SONOITA CREEK	AZ	3.00	1200

Consider:

PNO	PNAME	STATE	ACRES	FEE
7	MULESHOE RANCH	AZ	49120	0.00

Evaluate: WHERE ACRES > 1000 OR (STATE = 'AZ' AND FEE = 3.00)

TRUE OR (TRUE AND FALSE)

TRUE OR FALSE

TRUE

Sample Query 4.9: Display the PNAME, STATE, FEE, and ACRES values of any nature preserve with more than 1000 acres or is located in Arizona, and has a $3.00 admission fee (regardless of its location and acreage).

```
SELECT PNAME, STATE, FEE, ACRES
FROM   PRESERVE
WHERE (ACRES > 1000 OR STATE = 'AZ') AND FEE = 3.00
```

PNAME	STATE	FEE	ACRES
HASSAYAMPA RIVER	AZ	3.00	660
RAMSEY CANYON	AZ	3.00	380
PAPAGONIA-SONOITA CREEK	AZ	3.00	1200

Consider:

PNO	PNAME	STATE	ACRES	FEE
7	MULESHOE RANCH	AZ	49120	0.00

Evaluate: WHERE (ACRES > 1000 OR STATE = 'AZ') AND FEE = 3.00)

(TRUE OR TRUE) AND FALSE

TRUE AND FALSE =

FALSE

"Display Every Row Except…"

Sample Query 4.10: Display the PNO, STATE, and FEE values of every nature preserve *except* for those preserves that are located in Arizona and have a $3.00 admission fee.

```
SELECT PNO, STATE, FEE
FROM   PRESERVE
WHERE  NOT (STATE = 'AZ' AND FEE = 3.00 )
```

PNO	STATE	FEE
3	MT	0.00
7	AZ	0.00
40	MT	0.00
14	MA	0.00
13	MA	0.00
9	MA	0.00
11	MA	0.00
12	MA	0.00
1	MT	0.00
2	MT	0.00
10	MA	0.00

Hierarchy of Logical Operators

- NOTs are evaluated first
- ANDs are evaluated second
- ORs are evaluated third

Ex-1: WHERE ACRES > 1000 OR STATE = 'AZ' AND FEE = 3.00

WHERE ACRES > 1000 OR **(**STATE = 'AZ' AND FEE = 3.00)

Ex-2: WHERE NOT FEE = 3.00 OR ACRES > 1000 AND STATE = 'AZ'

WHERE (NOT FEE = 3.0) OR (ACRES > 1000 AND STATE = 'AZ')

An Ugly WHERE-Clause

Sample Query 4.11: Display PNAME, STATE, FEE, and ACRES values about all preserves that are smaller than 50 acres, or any preserve that has a $3.00 admission fee and is not located in Massachusetts.

```
SELECT PNAME, STATE, FEE, ACRES
FROM   PRESERVE
WHERE ACRES < 50
OR       FEE = 3.00
AND      NOT STATE = 'MA'
```

PNAME	STATE	FEE	ACRES
HASSAYAMPA RIVER	AZ	3.00	660
TATKON	MA	0.00	40
MIACOMET MOORS	MA	0.00	4
RAMSEY CANYON	AZ	3.00	380
PAPAGONIA-SONOITA CREEK	AZ	3.00	1200

```
WHERE  ACRES < 50
OR       (FEE = 3.00 AND (NOT STATE = 'MA'))
```

C. Logically Equivalent WHERE-Clauses

Trivial Example: WHERE NOT STATE = 'MA'

WHERE STATE <> 'MA'

SQ 4.8.

ACRES >1000 OR (STATE = 'AZ' AND FEE=3.00)

Equivalent to: (not immediately obvious)

(ACRES>1000 OR STATE = 'AZ') AND (ACRES>1000 OR FEE=3.00)

SQ 4.9:

(ACRES > 1000 OR STATE = 'AZ') AND FEE = 3.00

Equivalent to: (almost obvious)

FEE = 3.00 AND (ACRES > 1000 OR STATE = 'AZ')

Equivalent to: (not immediately obvious)

(FEE=3.00 AND ACRES > 1000) OR (FEE=3.00 AND STATE = 'AZ')

SQ 4.10: WHERE NOT (STATE = 'AZ' AND FEE = 3.00)

Equivalent to: (not immediately obvious)

WHERE (NOT STATE = 'AZ') OR (NOT FEE = 3.00)

Equivalent to: (obvious)

WHERE (STATE <> 'AZ') OR (FEE <> 3.00)

Four Interesting Questions

- How do you *really* know that two WHERE-clauses are equivalent?

  Logical Gymnastics (Laws of Logic)

- Does this "equivalent WHERE-clause stuff" really matter?

  Maybe – Someday you read another user's WHERE-clause

- Which WHERE-clause is friendlier? Intuition, Psychology

- Which WHERE-clause is more efficient? Query Optimization

”Logical Gymnastics” using Laws of Logic

Most users rarely have to play this game. [Author: In a typical SQL course, I do not enough time to present a comprehensive discussion of logic. I present basic logic, and “raise the anxiety level.”]

Assume Logical-Expression-1 is correct.

Convert: Logical-Expression-1 → Logical-Expression-2
or
Ask: Is Logical-Expression-1 *equivalent to* Logical-Expression-2 ?

How? *Laws of Logic (“Mechanical” “Cook Book”)”*

Laws of Logic

1. Distribute OR over AND

C1 OR (C2 AND C3) = (C1 OR C2) AND (C1 OR C3)

SQ 4.8:

ACRES>1000 OR (STATE = 'AZ' AND FE= 3.00)

(ACRES>1000 OR STATE = 'AZ') AND (ACRES>1000 OR FEE= 3.00)

2. Distribute AND over OR

C1 AND (C2 OR C3) = (C1 AND C2) OR (C1 AND C3)

SQ 4.9:

(ACRES > 1000 OR STATE = 'AZ') AND FEE=3.00

"Swap" AND-conditions

FEE=3.00 AND (ACRES > 1000 OR STATE = 'AZ')

Apply distributive law

(FEE=3.00 AND ACRES > 1000) OR (FEE=3.00 AND STATE = 'AZ')

3. De Morgan's Laws

1. NOT (C1 AND C2) = (NOT C1) OR (NOT C2)

2. NOT (C1 OR C2) = (NOT C1) AND (NOT C2)

SQ 4.10: NOT (STATE = 'AZ' AND FEE = 3.00)

By De Morgan: (NOT STATE = 'AZ') OR (NOT FEE = 3.00)

Optionally, rewrite as: (STATE <> 'AZ') OR (FEE <> 3.00)

Optionally, rewrite as: STATE <> 'AZ' OR FEE <> 3.00

Summary Suggestion: Learning Logic

How do you know that two WHERE-clauses are equivalent?

1. Do your best to “figure it out.”
2. Execute two SELECT statements with each WHERE-clause.
3. *** Logical gymnastics using laws of logic (Appendix 4B).

4. Truth Tables should produce same final result.

Truth Tables

SQ 4.10: WHERE NOT (STATE = 'AZ' AND FEE = 3.00)

WHERE NOT (C1 AND C2)

C1	C2	C1 AND C2	NOT (C1 AND C2)
T	T	T	F
T	F	F	T
F	T	F	T
F	F	F	T

C1	C2	NOT C1	NOT C2	(NOT C1) OR (NOT C2)
T	T	F	F	F
T	F	F	T	T
F	T	T	F	T
F	F	T	T	T

Summary: Articulating Query Objectives

Make sure that you really understand your query objectives. Writing a precise, concise, unambiguous query objectives in a human language can be a challenge.

Example: Display the PNO, PNAME, and STATE values of all preserves in Arizona *and* Montana.

```
SELECT PNO, PNAME, STATE
FROM    PRESERVE
WHERE  STATE = 'AZ' AND STATE = 'MT'
```

"No hit,"

Introduction to Appendices 4A - 4B - 4C

“Theory is Practical” - C. J. Date

Appendix 4A (Efficiency) - Query Optimization.

Appendix 4B (Theory) - Some Laws of Logic

Appendix 4C - Laws of Logic help optimizer

Next 3 Chapters/Videos

Chapter/Video 5: IN and BETWEEN

WHERE PNO IN (2, 9, 5)

.

Chapter/Video 6: LIKE

WHERE PNAME LIKE ('M%')

Chapter/Video 7: Arithmetic Expressions

SELECT FEE + 5.00

Future Chapters/Videos

Part II	Built-in Functions & Null Values
Chapter 8	Aggregate Functions
Chapter 9 -10	GROUP BY
Chapter 11	Null Values

Part III	Data Definition & Data Manipulation
Chapter 12	Preview Sample Sessions (All users)
Chapter 13-15	(Application Developers)

Part IV	Join Operations (Finally! Multiple tables)
Chapters 16–20	

.

.