3 Relational Algebra and QBE

Prerequisites for This Section

- **Readings:**
  - **Required:** Connolly and Begg, Section 2.2
  - **Required:** Connolly and Begg, section 4.1
  - **Required:** Connolly and Begg, sections 7.1 and 7.2
- **Assessments**
  - Multiple-Choice Quiz 2

Section objectives

In this section you will learn:

①The relational algebra is a theoretical language with operations that work on one or more relations to define another relation.

②Five basic operations in relational algebra: Selection, Projection, Cartesian product, Union, and Set Difference.

③Join, Intersection, and Division operations can be expressed in terms of five basic operations.

④How to form queries in relational algebra.

⑤The main features of Query-By-Example (QBE).

DreamHome Rental Database

The relational schema for part of DreamHome case study is:

- **Branch** (<u>branchNo</u>, street, city, postcode)
- **Staff** (<u>staffNo,</u> fName, lName, position, sex, DOB, salary, branchNo)
- **PropertyForRent** (<u>propertyNo</u>, street, city, postcode, type, rooms, rent, ownerNo, staffNo, branchNo)
- **Client** (<u>clientNo</u>, fName, lName, telNo, prefType, maxRent)
- **PrivateOwner** (<u>ownerNo</u>, fName, lName, address, telNo)
- **Viewing** (<u>clientNo</u>, <u>propertyNo</u>, viewDate, comment)
- **Registration** (<u>clientNo</u>, <u>branchNo</u>, <u>staffNo</u>, dateJoined)

Agenda

1. Relational Algebra
2. Set Operations
3. Native Relational Operations
4. The Interdependence of Operations
5. Illustrative Examples
6. Query-By-Example

- **Definition:**

  The relational algebra is a theoretical language with operations that work on one or more relations to define another relation without changing the original relation(s).

- **Property – closure:**

  - Both the operands and the results are relations, and so the output from one operation can become the input to another operation.
  - This allows expressions to be nested in the relational algebra, just as we can nest arithmetic operations.

The Operations in Relational Algebra

1. **Five basic operations**: Selection, Projection, Cartesian product, Union, and Set difference.
   - These perform most of the data retrieval operations needed.
2. **Three other operations:** Join, Intersection, and Division operations,
   - These can be expressed in terms of the five basic operations.

Two Types of Relational Algebra Operations

① Set operations

NAME	SYMBOL	KEYBOARD FORM	EXAMPLE
UNION	$\cup$	UNION	$R \cup S$, or R UNION S
INTERSECTION	$\cap$	INTERSECT	$R \cap S$, or R INTERSECT S
DIFFERENCE	$-$	$-$ or MINUS	$R - S$, or R MINUS S
CARTESION PRODUCT	$\times$	TIMES	$R \times S$, or R TIMES S

② Native relational operations

NAME	SYMBOL	KEYBOARD FORM	EXAMPLE
PROJECTION	$\Pi_{col1, \ldots, coln}(R)$	R[ ]	$R[A_{i1} \ldots A_{ik}]$
SELECTION	$\sigma_{predicate}(R)$	R where C	R where $A_1 = 5$
JOIN	$\bowtie$	JOIN	$R \bowtie S$, or R JOIN S
DIVISION	$\div$	DIVIDEBY	$R \div S$, or R DIVIDEBY S

Agenda

1. Relational Algebra
2. Set Operations
3. Native Relational Operations
4. The Interdependence of Operation
5. Illustrative Examples
6. Query-By-Example

Compatible Tables

- Tables are defined as sets of rows.

A	B	C
a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3

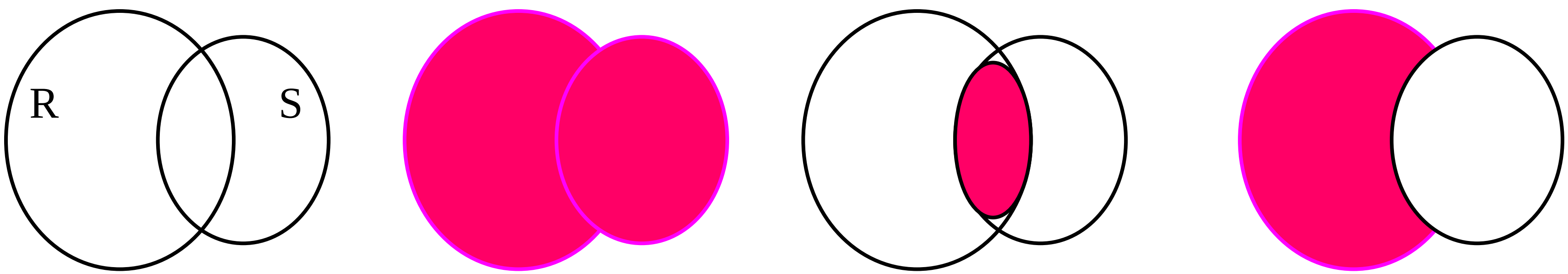
$\Longrightarrow$ $\{(a_1,b_1,c_1), (a_2,b_2,c_2), (a_3,b_3,c_3)\}$

- Tables R and S are *union-compatible (compatible)* if they have the same headings; that is, if Head(R) = Head(S), with attributes chosen from the same domains and with the same meanings.

- Only tables that are *union-compatible (compatible)* can be involved in unions, intersections, and set differences.

Venn Diagram

- The operations of intersection, union, and set difference are often pictured schematically using Venn diagrams, as illustrated below.



R∪S R∩S R−S

Union Operation

- $R \cup S$
  - The union of two relations R and S defines a relation that contains all the tuples of R, or S, or both R and S, duplicate tuples being eliminated.
  - R and S must be union-compatible.
- If R and S have *I* and *J* tuples, respectively, union is obtained by concatenating them into one relation with a maximum of (*I* + *J*) tuples.

Set Difference Operation

- R – S
  - Defines a relation consisting of the tuples that are in relation R, but not in S.
  - R and S must be union-compatible.

Intersection Operation

- $R \cap S$
  - Defines a relation consisting of the set of all tuples that are in both R and S.
  - R and S must be union-compatible.

Examples of ∪, ∩, and -

- Consider the tables R and S:

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_3
a_2	b_1	c_2

S

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_2	c_3
a_3	b_2	c_3

R∪S

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_2	c_3
a_2	b_1	c_2
a_3	b_2	c_3

R∩S

A	B	C
a_1	b_1	c_1
a_1	b_2	c_3

R-S

A	B	C
a_2	b_1	c_2

Cartesian Product Operation

- R X S
  - Defines a relation that is the concatenation of every tuple of relation R with every tuple of relation S.
  - The Cartesian product operation multiplies two relations to define another relation consisting of all possible pairs of tuples from the two relations.
  - If R has *I* tuples with *M* attributes and S has *J* tuples with *N* attributes, the R $\times$ S will contain (*I*J*) tuples with (*M+N*) attributes.

Example - Cartesian Product Operation

R

A	B	C
a_1	b_1	c_1
a_1	b_2	c_3
a_2	b_1	c_2

S

B	C	D
b_1	c_1	d_1
b_1	c_1	d_3
b_2	c_2	d_2
b_1	c_2	d_4

R×S

R.A	R.B	R.C	S.B	S.C	S.D
a_1	b_1	c_1	b_1	c_1	d_1
a_1	b_1	c_1	b_1	c_1	d_3
a_1	b_1	c_1	b_2	c_2	d_2
a_1	b_1	c_1	b_1	c_2	d_4
a_1	b_2	c_3	b_1	c_1	d_1
a_1	b_2	c_3	b_1	c_1	d_3
a_1	b_2	c_3	b_2	c_2	d_2
a_1	b_2	c_3	b_1	c_2	d_4
a_2	b_1	c_2	b_1	c_1	d_1
a_2	b_1	c_2	b_1	c_1	d_3
a_2	b_1	c_2	b_2	c_2	d_2
a_2	b_1	c_2	b_1	c_2	d_4

Agenda

1. **Relational Algebra**
2. **Set Operations**
3. **Native Relational Operations**
4. **The Interdependence of Operations**
5. **Illustrative Examples**
6. **Query-By-Example**

Selection (or Restriction) Operation

- $\sigma_{predicate}$ (R)
  - Works on **a single relation** R and defines a relation that contains only those tuples (rows) of R that satisfy the specified condition (*predicate*).
  - Is a unary operation.

Example - Selection Operation

- List all staff with a salary greater than £10,000.

$$\sigma_{salary > 10000} \text{ (Staff)}$$

staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24- Mar-58	18000	B003
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003

Projection Operation

- $\Pi_{\text{col1, . . . , coln}}(R)$
  - Works on **a single relation** R and defines a relation that contains a vertical subset of R, extracting the values of specified attributes and eliminating duplicates.
  - Is a unary operation.

Example – Projection Operation

- Produce a list of salaries for all staff, showing only staffNo, fName, lName, and salary details.

$$\Pi_{\text{staffNo, fName, lName, salary}}(\text{Staff})$$

staffNo	fName	lName	salary
SL21	John	White	30000
SG37	Ann	Beech	12000
SG14	David	Ford	18000
SA9	Mary	Howe	9000
SG5	Susan	Brand	24000
SL41	Julie	Lee	9000

Example - Cartesian Product and Selection

- List the names and comments of all clients who have viewed a property for rent.
  - Use selection operation to extract those tuples where Client.clientNo = Viewing.clientNo from the cartesian product result of Client and Viewing.

$$\sigma_{\text{Client.clientNo} = \text{Viewing.clientNo}}((\Pi_{\text{clientNo, fName, lName}}(\text{Client})) \text{ X } (\Pi_{\text{clientNo, propertyNo, comment}}(\text{Viewing})))$$

client.clientNo	fName	lName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR56	PG36	
CR62	Mary	Tregear	CR62	PA14	no dining room

Cartesian product and Selection can be reduced to a single operation called a *Join*.

Join Operation

- The Join operation is one of the essential operations in relational algebra.
- Join is a derivative of Cartesian product, equivalent to performing a Selection, using join predicate as selection formula, over Cartesian product of the two operand relations.
- One of the most difficult operations to implement efficiently in an RDBMS and one reason why RDBMSs have intrinsic performance problems.
- Join operation is a binary operation.

Example - Join Operation

R

A	B1	B2
a_1	b_1	b_1
a_1	b_2	b_1
a_2	b_1	b_2

S

B1	B2	C
b_1	b_1	c_1
b_1	b_1	c_2
b_1	b_2	c_3
b_2	b_2	c_4

R×S

A	R.B1	R.B2	S.B1	S.B2	C
a_1	b_1	b_1	b_1	b_1	c_1
a_1	b_1	b_1	b_1	b_1	c_2
a_1	b_1	b_1	b_1	b_2	c_3
a_1	b_1	b_1	b_2	b_2	c_4
a_1	b_2	b_1	b_1	b_1	c_1
a_1	b_2	b_1	b_1	b_1	c_2
a_1	b_2	b_1	b_1	b_2	c_3
a_1	b_2	b_1	b_2	b_2	c_4
a_2	b_1	b_2	b_1	b_1	c_1
a_2	b_1	b_2	b_1	b_1	c_2
a_2	b_1	b_2	b_1	b_2	c_3
a_2	b_1	b_2	b_2	b_2	c_4

R⋈S

A	R.B1	R.B2	C
a_1	b_1	b_1	c_1
a_1	b_1	b_1	c_2
a_2	b_1	b_2	c_3

Various Forms of Join Operation

① Theta join (θ-join)

② Equijoin (a particular type of Theta join)

③ Natural join

④ Outer join

① Theta join (θ-join)

- $R \bowtie_F S$
  - Defines a relation that contains tuples satisfying the predicate F from the Cartesian product of R and S.
  - The predicate F is of the form $R.a_i\ \theta\ S.b_i$ where θ may be one of the comparison operators ($<$, $\leq$, $>$, $\geq$, $=$, $\neq$).
  - Can rewrite Theta join using basic Selection and Cartesian product operations.

$$R \bowtie_F S = \sigma_F(R \times S)$$

② Equijoin

- In the case of Theta join (θ-join), if predicate F contains only equality (=), the term *Equijoin* is used.

② Example - Equijoin

- List the names and comments of all clients who have viewed a property for rent.

$$(\Pi_{\text{clientNo, fName, lName}}(\text{Client})) \bowtie_{\textbf{Client.clientNo = Viewing.clientNo}} (\Pi_{\text{clientNo, propertyNo, comment}}(\text{Viewing}))$$

client.clientNo	fName	lName	Viewing.clientNo	propertyNo	comment
CR76	John	Kay	CR76	PG4	too remote
CR56	Aline	Stewart	CR56	PA14	too small
CR56	Aline	Stewart	CR56	PG4	
CR56	Aline	Stewart	CR56	PG36	
CR62	Mary	Tregear	CR62	PA14	no dining room

③ Natural join

- $R \bowtie S$
  - An Equijoin of the two relations R and S over all common attributes *x*. One occurrence of each common attribute is eliminated from the result.

③ Example - Natural join

- List the names and comments of all clients who have viewed a property for rent.

$$(\Pi_{\text{clientNo, fName, lName}}(\text{Client})) \bowtie (\Pi_{\text{clientNo, propertyNo, comment}}(\text{Viewing}))$$

clientNo	fName	lName	propertyNo	comment
CR76	John	Kay	PG4	too remote
CR56	Aline	Stewart	PA14	too small
CR56	Aline	Stewart	PG4	
CR56	Aline	Stewart	PG36	
CR62	Mary	Tregear	PA14	no dining room

④ Outer join

- To display rows in the result that do not have matching values in the join column, use Outer join.

- R ⟕ S
  - Left outer join is join in which tuples from R that do not have matching values in common columns of S are also included in result relation.
  - Missing values in the S are set to null.

④ Example - Left Outer join

- Produce a status report on property viewings.

$$\Pi_{\text{propertyNo, street, city}}(\text{PropertyForRent}) ⟕ \text{Viewing}$$

propertyNo	street	city	clientNo	viewDate	comment
PA14	16 Holhead	Aberdeen	CR56	24-May-01	too small
PA14	16 Holhead	Aberdeen	CR62	14-May-01	no dining room
PL94	6 Argyll St	London	null	null	null
PG4	6 Lawrence St	Glasgow	CR76	20-Apr-01	too remote
PG4	6 Lawrence St	Glasgow	CR56	26-May-01	
PG36	2 Manor Rd	Glasgow	CR56	28-Apr-01	
PG21	18 Dale Rd	Glasgow	null	null	null
PG16	5 Novar Dr	Glasgow	null	null	null

Division Operation

- R ÷ S
  - Assume relation R is defined over the attribute set X and relation S is defined over the attributed set Y. Let Z = X-Y.
  - R ÷ S defines a relation over the attributes Z that consists of a set of tuples from R that match a combination of ***every*** tuple in S.

R

A	B	C
a_1	b_3	c_1
a_2	b_2	c_1
a_3	b_2	c_2
a_4	b_3	c_1

S

C
c_1

T=R÷S

A	B
a_1	b_3
a_2	b_2
a_4	b_3

Example – Division operation

- R ÷ S

R

A	B	C	D
a	b	c	d
a	b	e	f
b	c	e	f
e	d	c	d
e	d	e	f
a	b	d	e

÷

S

C	D
c	d
e	f

=

R÷S

A	B
a	b
e	d

Agenda

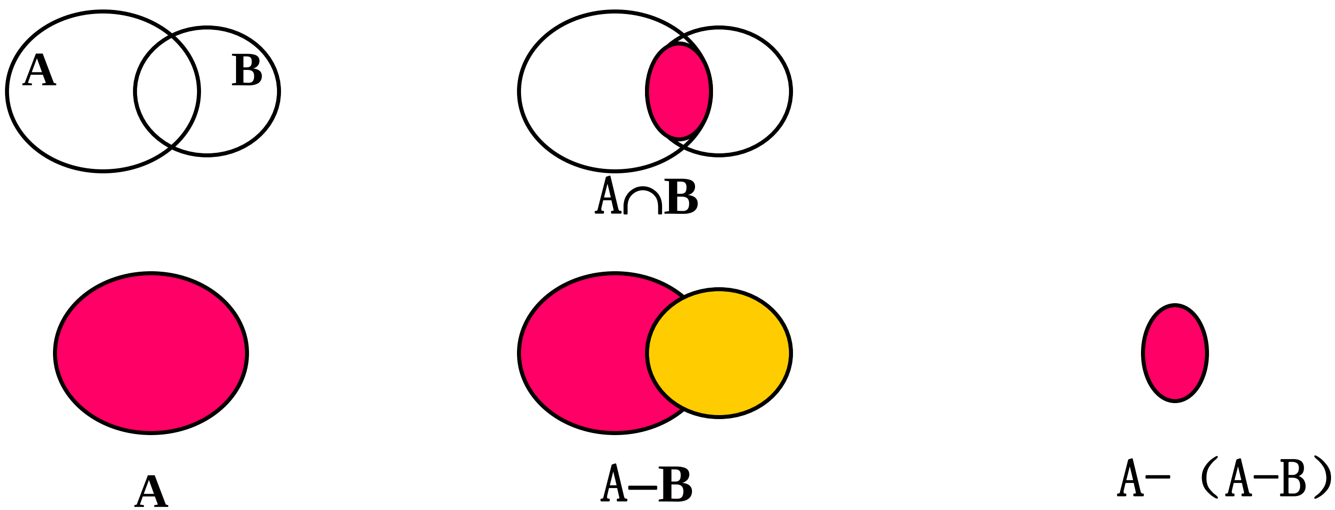
1. **Relational Algebra**
2. **Set Operations**
3. **Native Relational Operations**
4. **The Interdependence of Operations**
5. **Illustrative Examples**
6. **Query By Example**

Interdependence of Operations

- Relational completeness
  - The relational algebra is used to measure the selective power of relational languages.
  - A language that can be used to produce any relation that can be derived using the relational algebra is said to be relational complete.
- A minimal set of *basic operations* consists of union, set difference, cartesian product, selection, and projection.
- The remaining operations – intersection, join, and division – can be expressed using the basic operations.

Intersection Operation

- The Intersection operation can be defined in terms of Set difference alone.
  - $\mathbf{A \cap B = A - (A - B)}$
- Demonstrate by Venn diagrams



Join Operation

- The join of two tables R and S can be expressed using product, selection, and projection.

- $\mathbf{R \bowtie S = \Pi_{A,\, R.B,\, C}(\sigma_{R.B=S.B}(R \times S))}$

Division Operation

- Division can be expressed using projection, product, and difference.
  - Given relation R(A) and S(B), C = A - B
  - $$\mathbf{R \div S = \Pi_C(R) - \Pi_C((S \times (\Pi_C(R))) - R)}$$

Agenda

1. Relational Algebra
2. Set Operations
3. Native Relational Operations
4. The Interdependence of Operations
5. Illustrative Examples
6. Query By Example

Example 1- Union

- List all cities where there is either a branch office or a property for rent.

$$\Pi_{city}(\text{Branch}) \cup \Pi_{city}(\text{PropertyForRent})$$

city
London
Aberdeen
Glasgow
Bristol

Example 2 - Set Difference

- List all cities where there is a branch office but no properties for rent.

$$\Pi_{city}(\text{Branch}) - \Pi_{city}(\text{PropertyForRent})$$

city
Bristol

Example 3 - Intersection

- List all cities where there is both a branch office and at least one property for rent.

$$\Pi_{city}(\text{Branch}) \cap \Pi_{city}(\text{PropertyForRent})$$

city
Aberdeen
London
Glasgow

Example 4 – Join, Cartesian

- List the staff who work in the branch at '163 Main St.'.

$$\Pi_{\text{staffNo, fName, lName, position}}(\sigma_{\text{street='163 Main St.'}}(\text{Staff} \bowtie \text{Branch}))$$

$$\Pi_{\text{staffNo, fName, lName, position}}(\sigma_{\text{street ='163 Main St.'} \wedge \text{Staff.branchNo = Branch.branchNo}}(\text{Staff} \times \text{Branch}))$$

Example 5 - Division

- Identify all clients who have viewed all properties with three rooms.

$$(\Pi_{clientNo,\ propertyNo}(Viewing)) \div$$

$$(\Pi_{propertyNo}(\sigma_{rooms\ =\ 3}\ (PropertyForRent)))$$

$\Pi_{clientNo,propertyNo}(Viewing)$

clientNo	propertyNo
CR56	PA14
CR76	PG4
CR56	PG4
CR62	PA14
CR56	PG36

$\Pi_{propertyNo}(\sigma_{rooms=3}(PropertyForRent))$

propertyNo
PG4
PG36

RESULT

clientNo
CR56

Agenda

1. Relational Algebra
2. Set Operations
3. Native Relational Operations
4. The Interdependence of Operations
5. Illustrative Examples
6. Query-By-Example

Query-By-Example (QBE)

- Visual approach for accessing information in a database through use of query templates.
- Originally developed by IBM in 1970s and has proved so popular that QBE (or similar) is now provided by most DBMSs.
- When user constructs a QBE - in background, DBMS creates an equivalent SQL statement.

Properties of QBE

- QBE works as follows:
  - The system provides the user with a *skeleton* or *query template* of the tables in the database.
  - The user fills in the tables with examples of what are to be retrieved or updated.
- A *query template* of a table is a copy of the table without any rows, i.e. an empty table
- *Examples* can be:
  - Example variables or example elements (with underscore)
  - Constant values (are used to be select conditions)

Query Template of QBE

Relation Name	Attribute Name 1	Attribute Name 2	Attribute Name n

Query commands:

- Retrieved **P.**
- Update **U.**
- Insert **I.**
- Delete **D.**

- Example variables or example elements（with an underscore）
- Constant values

A Library Database Case Study

- The goal of the library database is to maintain detailed information about :1) library members, 2) titles included in the library's holdings, 3) borrowed books, and 4) hold requests.

① MEMBER()
② TITLE()
③ BOOK()
④ HOLD()

MEMBER	MemNo	DriverLic	Fname	M	Lname	Address	PhoneNumber
	PK						

BOOK	Book_Id	CallNumber	Edition	BorrowerMemNo	BorrowerDueDate
	PK			FK	

Projection in QBE

- QBE1: Displays MemNo, Lname, and PhoneNumber from MEMBER.

MEMBER	MemNo	DriverLic	Fname	M	Lname	Address	PhoneNumber
	P.				**P.**		**P.**

Selection in QBE

- QBE2: Retrieve all members whose first name is John.

MEMBER	MemNo	DriverLic	Fname	M	Lname	Address	PhoneNumber
P.			**John**				

- By placing P. under the table name, this will retrieve and display the data in all the columns.

Selection in QBE

- QBE3: Retrieve the name and member number of all the members whose member number is greater than 100.

MEMBER |MemNo | DriverLic| Fname| M| Lname| Address| PhoneNumber|

P. >100 **P.** **P.**

- Comparison with constant value (in the above example the constant value is 100) is placed in the appropriate column.

- The resulting table will have the following columns:

Result| MemNo | Fname | Lname |

Selection in QBE

- In QBE, a disjunction (OR) is expressed by using different examples in different rows of the skeleton.

- QBE4: Retrieve the name and member number of all the members whose first name is John or Susan.

MEMBER	MemNo	DriverLic	Fname	M	Lname	Address	PhoneNumber
	P.		**P.John**		**P.**		
	P.		**P.Susan**		**P.**		

Selection in QBE

- A conjunction (AND), on the other hand, is expressed in the same row.

- QBE5: Retrieve the name and member number of all the members whose first name is Susan and whose member number is greater than 100.

MEMBER	MemNo	DriverLic	Fname	M	Lname	Address	PhoneNumber
	P.>100		**P.Susan**		**P.**		

- If the conjunction is a condition involving a single column, the condition can be specified using the AND operator, as in SQL. For example, if the MemNo should be greater than 100 and less than 150, this is specified under the MemNo column as: (<u>x</u> > 100) AND (<u>x</u> < 150)

Join in QBE

- Joins can be expressed by using common example variables in multiple tables in the columns to be joined.
- QBE6: List the member number and last name of all the members who currently have a borrowed book.

MEMBER	MemNo	DriverLic	Fname	M	Lname	Address	PhoneNumber
	P.join				**P.**		

BOOK	Book_id	CallNumber	Edition	BorrowerMemNo	BorrowDueDate
				join	

QBE Allows User to

① Ask questions about data in one or more tables

② Specify the fields we want in the answer.

③ Select records according to some criteria.

④ Perform calculations on the data in tables.

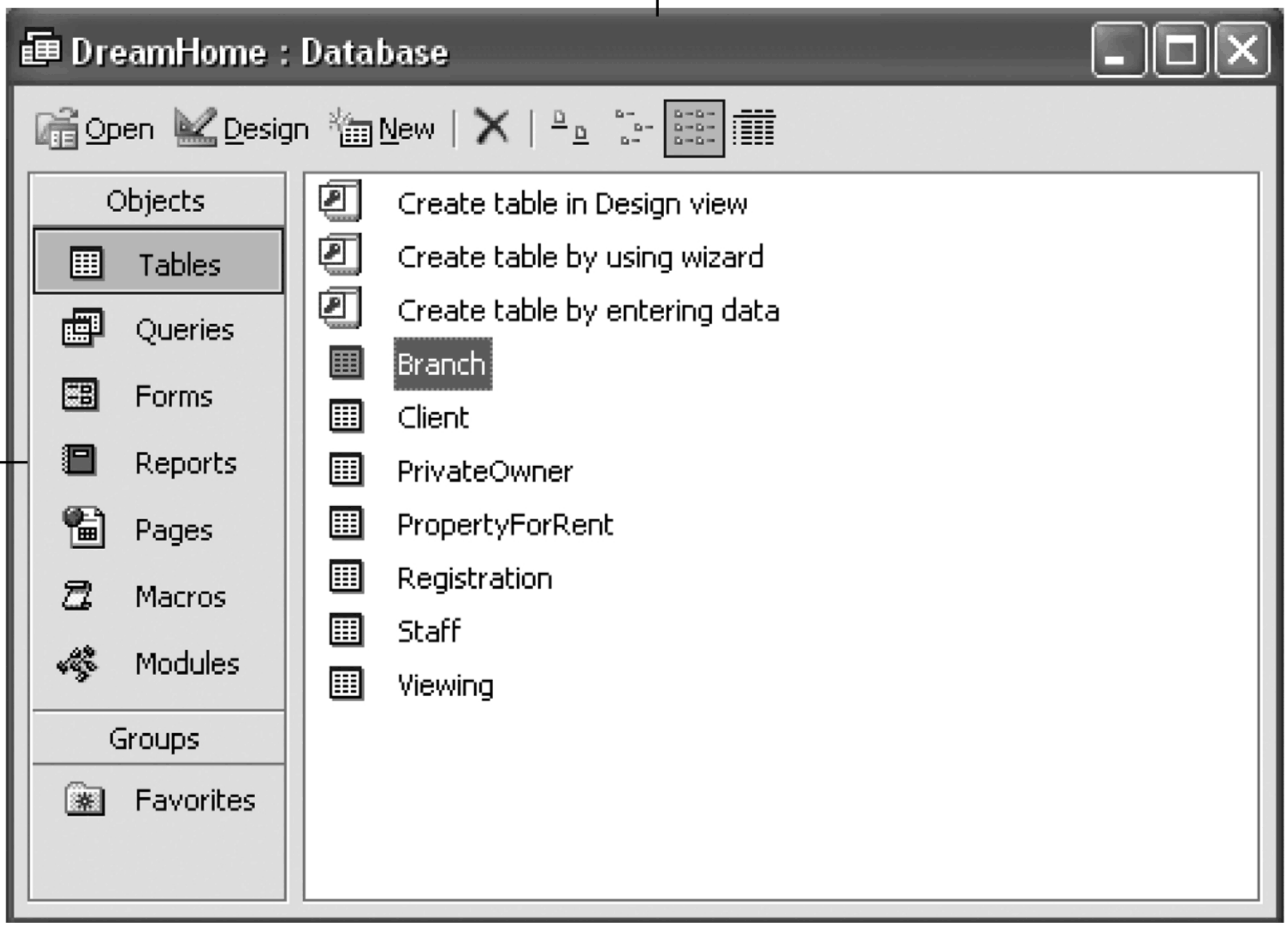
⑤ Insert and delete records.

⑥ Modify values of fields.

⑦ Create new fields and tables.

Introduction to Microsoft Access

Database window



Database objects

Section objectives

In this section you will learn:

① The relational algebra is a theoretical language with operations that work on one or more relations to define another relation.

② Five basic operations in relational algebra: Selection, Projection, Cartesian product, Union, and Set Difference.

③ Join, Intersection, and Division operations can be expressed in terms of five basic operations.

④ How to form queries in relational algebra.

⑤ The main features of Query-By-Example (QBE).

Questions?

Assignments

- Multiple-Choice Quiz 2
- Exercise 2

Prerequisites for Next Section

- **Readings:**
  - **Required:** Connolly and Begg, sections 5.1 and 5.2
  - **Required:** Connolly and Begg, sections 6.1, 6.2 and 6.3

- **Assessments:**
  - Multiple-Choice Quiz 3