Objective: Developing TQP queries

Answer the following queries in SQL, RA, and QBE by using OES2 database. For SQL and RA, if the query requires join then develop solution for TIMES, Equi-Join, and Natural Join

Note: Symbols to be used in RA: σ ⋈ π

**Query 1: List unique pair of customer\_no that both customers are from the same city.**

**--Relational Algebra (Time)**

C1 ⇐ C

π C.CUSTOMER\_NO, C1.CUSTOMER\_NO (σ C.CUSTOMER\_NO > C1.CUSTOMER\_NO  C.CITY = C1.CITY (C1) X (C))

**--Relational Algebra (Join)**

C1 ⇐ C

π C.CUSTOMER\_NO, C1.CUSTOMER\_NO (σ C.CUSTOMER\_NO > C1.CUSTOMER\_NO (C1) ⋈ C1.CITY = C.CITY (C))

**--Relational Algebra (Natural Join)**

C1 ⇐ C

π C.CUSTOMER\_NO, C1.CUSTOMER\_NO (**(C)** ⋈ C.CITY = C1.CITY  (σ C.CUSTOMER\_NO > C1.CUSTOMER\_NO **(C1)((C)** \*(C1)**)**))

**--Structure Query Language (Time)**

SELECT C.CUSTOMER\_NO, C1.CUSTOMER\_NO FROM CUSTOMER C,CUSTOMER C1

WHERE C.CITY = C1.CITY AND C.CUSTOMER\_NO > C1.CUSTOMER\_NO;

**--Structure Query Language (Join)**

SELECT C.CUSTOMER\_NO, C1.CUSTOMER\_NO FROM CUSTOMER C

JOIN CUSTOMER C1 ON C.CITY = C1.CITY

WHERE C.CUSTOMER\_NO > C1.CUSTOMER\_NO;

**--Structure Query Language (Natural Join)**

SELECT C.CUSTOMER\_NO, C1.CUSTOMER\_NO FROM CUSTOMER C

NATURAL JOIN CUSTOMER C1 USING (CITY)

WHERE C.CUSTOMER\_NO > C1.CUSTOMER\_NO;

**--QBE**

|  |  |  |  |
| --- | --- | --- | --- |
| CUSTOMER | CUSTOMER\_NO | **City** | Condition |
|  | P.\_CUSTOMER\_NOX | \_CITYX | \_CUSTOMER\_NOX >\_\_CUSTOMER\_NOX |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| CUSTOMER | CUSTOMER\_NO1 | **City1** |
|  | P.\_CUSTOMER\_NOX | \_CITYX |
|  |  |  |

--**RESULTS**

CUSTOMER\_NO CUSTOMER\_NO

--------------------------------------- ---------------------------------------

1110 1000

1025 1000

1003 1000

1119 1023

1111 1023

1120 1060

1112 1060

1115 1064

1113 1064

1073 1064

1072 1064

**Query 2: List customer last name, and customer\_no for customers who have purchased products with total order amount greater than $100,000.**

**--Relational Algebra (Time)**

π C.CUSTOMER\_NO, C.LNAME (σ O.TOTAL\_AMT > 100000 ∩ C.BRANCH\_NO = O.BRANCH\_NO (C) X (O))

**--Relational Algebra (Join)**

π C.CUSTOMER\_NO, C.LNAME(**(C)**⋈ C.BRANCH\_NO = O.BRANCH\_NO (σ O.TOTAL\_AMT > 100000 **(O)))**

**--Relational Algebra (Natural Join)**

π C.CUSTOMER\_NO, C.LNAME(**(C)** \* (π BRANCH\_NO (σ O.TOTAL\_AMT > 100000 **(O)** )))

**--Structure Query Language (Time)**

SELECT C.CUSTOMER\_NO, C.LNAME FROM CUSTOMER C, ORDERS O

WHERE O.TOTAL\_AMT > 100000;

**--Structure Query Language (Join)**

SELECT C.CUSTOMER\_NO, C.LNAME FROM CUSTOMER C

JOIN ORDERS O ON O.BRANCH\_NO = C. BRANCH\_NO

WHERE O.TOTAL\_AMT > 100000;

**--Structure Query Language (Natural Join)**

SELECT C.CUSTOMER\_NO, C.LNAME FROM CUSTOMER C

NATURAL JOIN ORDERS O USING (BRANCH\_NO)

WHERE O.TOTAL\_AMT > 100000;

**--QBE**

|  |  |  |  |
| --- | --- | --- | --- |
| **CUSTOMER** | **CUSTOMER\_NO** | **LNAME** | **BRANCH\_NO** |
|  | P. | P. | \_BRANCH\_NOX |

|  |  |  |
| --- | --- | --- |
| **BRANCH** | **BRANCH\_NO** | **TOTAL\_AMT** |
|  | \_BRANCH\_NOX | >10000 |

**--RESULTS**

CUSTOMER\_NO LNAME

1000 Jake

1023 Maria

1060 Doering

1064 Jenkins

1068 Halle

1072 Bond

1076 Thomas

1001 Case

**Query 3: List product description and price of the least expensive product.**

**--Relational Algebra**

π PRODUCT\_DESCRIPTION, UNIT\_PRICE (σ UNIT\_PRICE = (σ MIN(UNIT\_PRICE (P)))

**--Structure Query Language**

SELECT PRODUCT\_DESCRIPTION, UNIT\_PRICE FROM PRODUCT

WHERE UNIT\_PRICE = (SELECT MIN(UNIT\_PRICE) FROM PRODUCT);

**--QBE**

|  |  |  |  |
| --- | --- | --- | --- |
| PRODUCT | PRODUCT\_DESCRIPTION | UNIT\_PRICE | **CONDITION** |
|  | P. | P. | MIN(UNIT\_PRICE) |

**--RESULTS**

PRODUCT\_DESCRIPTION UNIT\_PRICE

Xbox Memeory Card 50

Mario Game for Xbox 50

Optical Desktop with Fingerprint Reader 50

**Query 4: List customer\_no, customer last name for those customers from Mankato Minnesota.**

**--Relational Algebra**

π CUSTOMER\_NO, LNAME (σ CITY = 'Mankato' ∩ STATE = 'Minnesota' (C))

**--Structure Query Language**

SELECT CUSTOMER\_NO, LNAME FROM CUSTOMER

WHERE CITY = 'Mankato' and STATE = 'Minnesota';

**--QBE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CUSTOMER** | **CUSTOMER\_NO** | **LNAME** | **CITY** | **STATE** |
|  | P. | P. | ‘Mankato’ | ‘Minnesota’ |

**--RESULTS**

CUSTOMER\_NO LNAME

1000 Jake

1003 Aslam

1025 George

1110 Hagen

**Query 5: List employee\_no and last name of those emploees that their job title is customer service.**

**--Relational Algebra (Time)**

π E.EMPLOYEE\_NO, E.LNAME (σ EP.DESCRIPTION = 'Customer Service'∩ E.POSITION = EP.POSITION (E) X (EP))

**--Relational Algebra (Join)**

π E.EMPLOYEE\_NO, E.LNAME (**(E)**⋈ E.POSITION = EP.POSITION (σ EP.DESCRIPTION = 'Customer Service' **(EP)))**

**--Relational Algebra (Natural Join)**

π E.EMPLOYEE\_NO, E.LNAME (**(E)** \* (σEP.DESCRIPTION = 'Customer Service' **(EP)**))

**--Structure Query Language (Time)**

SELECT E.EMPLOYEE\_NO, E.LNAME FROM EMPLOYEE E, EMPLOYEE\_POSITION EP

WHERE EP.DESCRIPTION = 'Custome Service';

**--Structure Query Language (Join)**

SELECT E.EMPLOYEE\_NO, E.LNAME FROM EMPLOYEE E

JOIN EMPLOYEE\_POSITION EP ON EP.POSITION = E.POSITION

WHERE EP.DESCRIPTION = 'Customer Service';

**--Structure Query Language (Natural Join)**

SELECT E.EMPLOYEE\_NO, E.LNAME FROM EMPLOYEE E

NATURAL JOIN EMPLOYEE\_POSITION EP USING (POSITION)

WHERE EP.DESCRIPTION = 'Customer Service';

**--QBE**

|  |  |  |  |
| --- | --- | --- | --- |
| **EMPLOYEE** | **EMPLOYEE \_NO** | **LNAME** | **POSITION** |
|  | P. | P. | **\_POSITIONX** |

|  |  |  |
| --- | --- | --- |
| **EMPLOYEE\_POSITION** | **POSITION** | **DESCRIPTION** |
|  | **\_POSITIONX** | 'Customer Service' |

**--RESULTS**

EMPLOYEE\_NO LNAME

1000 Wyatt

1001 Wright

1002 Worral

1003 Wooton

1004 Widdes

1005 Wehland

1006 Thomas

1007 Thomas

1008 Stone

1009 Stewart

1010 Stansbury

1011 Stansbury

1012 Somers

1013 Simmins