(Select update account set balance = (5 * balance)/100
balance where balance > (Select avg(balance) from account)

Ques. Delete the records of all a/c with balnc below the avg of the bank
Delete from account where balance < (select avg(balnc) from account)

Ques. Pay 5% interest on account whose bal > avg (above) (bal

5/09/17

* Tuple Relational Calculas:

It is non-procedural query language where each query is of the form
$\{t \mid P(t)\}$

- It represents the set of all tuple 't' such that the predicate P is true for all 't'.
  (condⁿ like in SQL)
- 't' is a tuple variable, t[A] denotes the value of tuple t on attribute A
- $t \in r$ denotes that 't' is in relⁿ 'r'.
- P is a formula similar to that of the predicate calculas.

* Predicate Calculas Formula:

(i) Set of attributes & constants
(ii). Set of comparison operators ($<, \leq, =, \neq, >, \geq$)
(iii). Set of connectives: and ($\wedge$), or ($\vee$), not ($\neg$)
(iv). Implication ($\Rightarrow$) $x \Rightarrow y$, if x is true then y is true
$x \Rightarrow y \equiv \neg x \vee y$
(v) Set of quantifiers
$\exists t \in r (Q(t))$ ≡ "there exist" a tuple 't' in in relⁿ r such that predicate Q is true
$\forall t \in r (Q(t))$ = Q is true for all "tuples t in 'r'

Teacher's Sign.

A tuple-relational calculus formula is build up out of atoms. An atom has one of the following forms:

- $s \in r$, where s is tuple variable & r is a rel^n (we do not allow use of the $\notin$ operator)
- $s[x] \; \theta \; u[y]$, where s and u are tuple variables, x is attribute on which s is defined, y is an attribute on which y is defined, & 'θ' is a comparison operator ($<, \leq, =, \neq, >, \geq$), we require that attribute x & y have domain whose member can be compared by θ
- $s[x] \; \theta \; c$, c is the constant in the domain of attribute x.

We build up formulae from atoms by using the following rules:

- An atom is a formula
- If P_1 is a formula, then so are $\neg P_1$ and (P_1)
- If P_1 & P_2 are formulae, then so are $P_1 \vee P_2$, $P_1 \wedge P_2$, $P_2 \Rightarrow P_1$
- If $P_1(s)$ is a formula containing a free tuple variables, s and r is a rel^n then

$\exists \; s \in r (P_1(s))$ and $\forall \; s \in r (P_1(s))$ are also formula

In the tuple relational calculus, equivalences include following three rules.

1. $P_1 \wedge P_2$ is equivalent to $\neg (\neg (P_1) \vee \neg (P_2))$
2. $\forall \; t \in r (P_1(t))$ is equivalent to $\neg \exists \; t \in r (\neg P_1(t))$
3. $P_1 \Rightarrow P_2$ is equivalent to $\neg (P_1) \vee P_2$

E.g Banking Example

Find the loan_no, br_name & amount for loan of over $1200

$$\{ t \mid t \in loan \wedge t[amount] > 1200 \}$$

Suppose that we want only the loan_no, rather than all attributes of loan relation

$$\{ t \mid \exists \; s \in loan (t[loan\_no] = s[loan\_no] \wedge s[amt] > 1200 \}$$

Find the name of all customer having a loan, an account, or both at the bank.

$$\{ t \mid \exists \; s \in borrower (t[customer\_name] = s[cust\_name]) \vee \exists \; u \in depositor (t[cust\_name] = u[cust\_name]) \}$$

6/09/17

Find the name of all customers who have a loan & an account at the bank

$$\{ t \mid \exists \; s \in borrower (t[customer\_name] = s[cust\_name]) \wedge \exists \; u \in depositor (t[cust\_name] = u[cust\_name]) \}$$

Find the name of all customers having a loan at perryridge Branch.

Select cust_name from Borrower where loan_no in (Select loan_no from loan where branch_name = 'perryridge')

$\{t \mid \exists s \in borrower (t[cust\_name] = s[cust\_name] \wedge \exists u \in loan (u[branch\_name] = \text{"Perryridge"} \wedge u[loan\_no] = s[loan\_no]))\}$

Find the names of all customers who have a loan at the perryridge branch but not acc at any branch of the bank

$\{t \mid \exists s \in borrower (t[cust\_name] = s[cust\_name] \wedge \exists u \in loan (u[branch\_name] = \text{"Perryridge"} \wedge u[loan\_no] = s[loan\_no])) \wedge \text{not } \exists v \in depositer (v[cust\_name] = t[cust\_name])\}$

Find all customers who have a/c at the bank to not have loan from the bank

$\exists s \in$ ~~borrower~~ depositor $(t[cust\_name] = s[cust\_name] \wedge \text{not } \exists u \in borrower (t[cust\_name] = s[cust\_name])$

Teacher's Sign.

8/09/17

Ques. Find the name of all customers having a loan from the perryridge branch & cities in which they live

$\{t \mid \exists s \in borrower (t[cust\_name] = s[cust\_name] \wedge \exists u \in loan (u[branch\_name] = \text{"perryridge"} \wedge u[loan\_no] = s[loan\_no])) \wedge \exists v \in customer (v[cust\_name] = t[cust\_name] \wedge (v[city] = v[city]))\}$

* Domain Relational Calculas:

It is a second form of relational calculas which uses domain variables that take on values from an attribute domain, rather than values for an entiere tuple. It is also non-procedural query language like tuple relational calculas. Each query is an expression of the form,

$\{\langle x_1, x_2, \ldots x_n \rangle \mid P(x_1, x_2, \ldots x_n)\}$

$\langle x_1, x_2, \ldots x_n \rangle$: Set of attribute (domain Variables)

$P(x_1, x_2, \ldots x_n)$: Predicate formula composed of atoms.

An atom in domain relational calculas has following forms-

- $\langle x_1, x_2, \ldots x_n \rangle \in r$, where r is a relation on n attributes & $x_1, x_2, \ldots x_n$ are domain variables or domain constants.
- $x \, \theta \, y$ where x & y are domain variables & θ is a comparison operator $(<, \leq, =, \neq, \geq, >)$. We require the attributes x & y have domain that can be compared by θ

• $x \theta c$, where x is domain variable, θ is comparison operator & c is constant in the domain of attribute x

We build up formulae from atoms by using following rules-

- An atom is formula
- If P_1 is formula, then so are $\neg P_1$ & (P_1)
- If P_1 & P_2 are formulae, then so are $P_1 \wedge P_2$, $P_1 \vee P_2$ & $P_1 \Rightarrow P_2$
- If $P_1(x)$ is formula in x; where x is domain variable. then,

  $\exists x (P_1(x))$ and $\forall x (P_1(x))$ are also formulae.
- As a notational shorthand, we write

  $\exists a, b, c (P(a, b, c,))$

  for

  $\exists a (\exists b (\exists c (P(a, b, c))))$

Ques. Find the loan_no, br_name & amount for loans over $1200

$\{ \langle loan\_no, br\_name, amt \rangle \mid \langle loan\_no, br\_name, amt \rangle \in loan \wedge amt > 1200 \}$

Ques. Find the name of all customers who have a loan of over $1200

$\exists$ loan_no, br_name, amt

loan_no

$\{ \langle cust\_name \rangle \mid \langle cust\_name, loan\_no \rangle \in borrower \wedge \langle loan\_no, br\_name, amt \rangle \in loan \wedge amt > 1200 \}$

Teacher's Sign

Ques. Find the name of all cust having loan from perryridge branch & the loan amount

$\{ \langle c, a \rangle \mid \exists l (\langle c, l \rangle \in borrower \wedge \exists b (\langle l, b, a \rangle \in loan \wedge b = \text{"Perryridge"})) \}$

OR

$\{ \langle c, a \rangle \mid \exists l (\langle c, l \rangle \in borrower \wedge \langle l, \text{"Perryridge"}, a \rangle \in loan) \}$

Ques. Find the name of all customers having a loan, an account or both at perryridge branch

Wrong

$\{ \langle c \rangle \mid \exists l (\langle c, l \rangle \in borrower \vee \exists ac \langle c, ac \rangle \in depositor) \wedge \exists b, bal (\langle ac, b, bal \rangle \in loan \wedge \langle l, b, bal \rangle \in account \wedge b = \text{"perryridge"}) \}$

$\{ \langle c \rangle \mid \exists l (\langle c, l \rangle \in borrower \wedge \exists b, a (\langle l, b, a \rangle \in loan \wedge b = \text{"perryridge"}) \vee \exists a (\langle c, a \rangle \in depositor \wedge \exists b, n (\langle a, b, n \rangle \in account \wedge b = \text{"perryridge"}))) \}$

Date / /
Page No

ence b/w DBMS & RDBMS:

DBMS	RDBMS
the concept of ship is missing. ist, it is very	• It is based on the concepts of relationship
f operation is very	• Speed of opⁿ is fast.
re & s/w requirement	• H/w and s/w requirements are very high.
& utilities offered are	• Facilities & utilities offered are many
concept of file	• It uses concept of table
ally use 3GL	• It normally use 4GL
ASE, FOXBASE, etc	E.g. ORACLE, INGRES

Popular RDBMS Packages -

RDBMS Package	Company / corporation
ORACLE	Oracle corporation
SYBASE	Sybase
INFORMIX	Informix software
(4). MYSQL	It is an open source.
(5). DB2	IBM
(6). INGRES	Computer Associates International
(7) SQL Server	Microsoft

Teacher's Sign.

limitation for some context.

Date / /
Page No

Update opⁿ in Relational Algebra & in SQL ??

Write the SQL datatypes.

Characteristics of SQL & its advantages.

* Normalization :

E-R modelling is used by Designer Analysis tool. Database designed based on ER model may contain some amount of inconsistency, ambiguity & redundancy. To resolve these issues some amt of refinement is required. The refinement process of db design is referred as Normalization.

As normalization invokes building structure (like tables) starting from the stage of identifying the columns (attributes) associated in the table, it is also called bottom up approach.

Basically, normalization eliminates the duplicate data & make insert, update & delete opⁿ much more efficient in terms of performance & space requirement to store the data.

Anomaly : Insertion anomaly, Updation anomaly, deletion anomaly.