

More on Oracle

Introduction / Aim

This exercise involves extending the database you worked on last week. Currently it has details about bank accounts and their owners, branches and employees. There are two parts to this lab. The first task is to extend the schema in order to hold information about transactions. The second is to execute the queries in SQL listed below. You could do question 4 before 1-3 if you like.

Tasks

Task A

1. A transaction is the payment of a sum of money from an account. How will we add this to the database? One answer is to create a table linked to the account. Create this table either as a strong entity table with a system-wide unique transaction number to act as a primary key, or as a weak entity type with this time the transaction number only being unique within the transaction – i.e. each account has a first transaction, a second one, etc. The transaction should also have the sum, the date, the payee (i.e. who or what the money is paid to) and which customer authorised the transaction.

```
CREATE TABLE Transaction(  
  number Number(6) CONSTRAINT pk_Transaction PRIMARY KEY,  
  acc Number(5) CONSTRAINT fk_tranacc REFERENCES Account(accountNo)  
    CONSTRAINT nn_acc NOT NULL,  
  sum Number(20,2) CONSTRAINT nn_sum NOT NULL,  
  date DATE CONSTRAINT nn_date NOT NULL,  
  payee VARCHAR2(50) CONSTRAINT nn_payee NOT NULL,  
  authorisedBy NUMBER(3) CONSTRAINT fk_author REFERENCES  
    Customer(id) )
```

Notes:

Number would probably use and autosequence

Make sure foreign key columns match the type – e.g. acc must be Number(5)

2. Enter a little data into the table.
3. Consider the following questions and give a general account of what the database system must allow us to do to answer them.

- a) Entering a transaction into the table doesn't actually do anything. What we need to do to keep the database consistent?

The mechanism for entering a new transaction should also update the balance of the account.

- b) How can we work out how much has been withdrawn from the account?

By a query which finds all the withdrawals from the account and then adds them up – using the sum aggregate function to be covered soon.

- c) How could we check that the person authorising the cheque was an owner of the account?

We need a much more powerful kind of constraint that states that the Customer record referred to in authorisedBy field is one of the customer records linked to the account through the Owner table. This much more complex constraint is called an Assertion.

- d) How could we make the database cope with both ends of a transaction – i.e. having the money withdrawn from one account?

Payee should be a foreign key referring to an account number.

- e) If this was a database for one bank how could we deal with transactions involving other banks?

We need to extend the database with a table for external accounts and allow the payee to refer to an entry in that table.

Task B

Note (for questions from (f) onwards) – if table A has a foreign key, FK, to the primary key of table B, PK, and you want the value of column X in table B for records which match records in table A where column Y = 1, you do:

```
SELECT X FROM A, B WHERE Y=1 AND FK=PK
```

4. Retrieve the following information from the original database.

- a) The set of branch numbers.

```
SELECT branchNo AS 'Branch Number' FROM Branch
```

- b) The set of customer names.

```
SELECT forename, surname FROM Customer
```

- c) The address of branch 3.

```
SELECT brAddress FROM Branch WHERE branchNo = 3
```

- d) The name of customer 198.

```
SELECT forename, surname  
FROM Customer WHERE id = 198
```

- e) The names of the employees of branch 3.

```
SELECT foreName, surname  
FROM Employee WHERE empBranch = 3
```

f) The name(s) of the owners of account 23520.

```
SELECT forename, surname  
FROM Customer, Owner WHERE accNo = 23520 AND custId = ID
```

g) The address of the branch that Rosemary Hodge works in.

```
SELECT brAddress FROM Branch, Employee  
WHERE empBranch = branchNo AND forename = 'Rosemary'  
AND surname = 'Hodge'
```

h) The balance(s) of the account(s) owned by Brian Carmichael.

```
SELECT balance FROM Account, Owner, Customer  
WHERE accountNo = accNo AND custId = ID AND forename = 'Brian'  
AND surname = 'Carmichael'
```

5. Retrieve the following information using the new Transaction table.

i) The sums withdrawn from account 23501 – note you should enter some transactions withdrawing money from this account.

```
SELECT sum  
FROM Transaction WHERE acc = 23501
```

j) The names of anyone who authorised a transaction in account 23503.

```
SELECT forename, surname  
FROM Customer, Transaction WHERE acc = 23503  
AND custId = authorisedBy
```

k) List the sums withdrawn from accounts in branch 4.

```
SELECT sum FROM Transaction, Account  
WHERE accountNo = acc AND inBranch = 4
```