Database Applications Laboratory

Subject Code: 06CSL57 **I.A. Marks: 25**
Hours/Week: 03 **Exam Hours: 03**
Total Hours: 42 **Exam Marks: 50**

I. Consider the Insurance database given below. The primary keys are underlined and the data types are specified.

PERSON (DRIVER_ID#: STRING, NAME: STRING, ADDRESS: STRONG)

CAR (REGNO: STRING, MODEL: STRING, YEAR: INT)

ACCIDENT (REPORT_NO: INT, ADATE: DATE, LOCATION: STRING)

OWNS (DRIVER_ID #: STRING. REGNO: STRING)

PARTICIPATED(DRIVER_ID#: STRING, REGNO: STRING, REPORT_NO:INT, DAMAGE_AMT: INT)

(i) Create the above tables by properly specifying the primary keys and the foreign keys.
(ii) Enter at least five tuples for each relation.
(iii) Demonstrate how you
a. Update the damage amount for the car with a specific Regno in the accident with report number 12 to 25000.
b. Add a new accident to the database.
(iv) Find the total number of people who owned cars that were involved in accidents in 2008.
(v) Find the number of accidents in which cars belonging to a specific model were involved.
(vi) Generate suitable reports.
(vii) Create suitable front end for querying and displaying the results.

II. Consider the following relations for an order processing database application in a company.

CUSTOMER (CUST #: INT, CNAME: STRING, CITY: STRING)

ORDER (ORDER #: INT, ODATE: DATE, CUST #: INT, ORD-AMT: INT)

ITEM (ITEM #: INT, UNIT PRICE: INT)

ORDER - ITEM (ORDER #: INT, ITEM #: INT, QTY: INT)

SHIPMENT (ORDER #: INT, WAREHOUSE#: INT, SHIP-DATE: DATE)

WAREHOUSE (WAREHOUSE #: INT, CITY: STRING)

(i) Create the above tables by properly specifying the primary keys and the foreign keys.
(ii) Enter at least five tuples for each relation.
(iii) Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column is the total numbers of orders by the customer and the last column is the average order amount for that customer.
(iv) List the order# for orders that were shipped from *all* the warehouses that the company has in a specific city.
(v) Demonstrate how you delete item# 10 from the ITEM table and make that field *null* in the ORDER_ITEM table.
(vi) Generate suitable reports.
(vii) Create suitable front end for querying and displaying the results.

III. Consider the following database of student enrollment in courses & books adopted for each course.

STUDENT (regno: string, name: string, major: string, bdate:date)
COURSE (course #:int, cname:string, dept:string)
ENROLL (regno:string, course#:int, sem:int, marks:int)
BOOK _ ADOPTION (course# :int, sem:int, book-ISBN:int)
TEXT (book-ISBN:int, book-title:string, publisher:string, author:string)

(i) Create the above tables by properly specifying the primary keys and the foreign keys.
(ii) Enter at least five tuples for each relation.
(iii) Demonstrate how you add a new text book to the database and make this book be adopted by some department.
(iv) Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.
(v) List any department that has *all* its adopted books published by a specific publisher.
(vi) Generate suitable reports.
(vii) Create suitable front end for querying and displaying the results.

IV. The following tables are maintained by a book dealer.

AUTHOR (author-id:int, name:string, city:string, country:string)
PUBLISHER (publisher-id:int, name:string, city:string, country:string)
CATALOG (book-id:int, title:string, author-id:int, publisher-id:int, category-id:int, year:int, price:int)
CATEGORY (category-id:int, description:string)
ORDER-DETAILS (order-no:int, book-id:int, quantity:int)

(i) Create the above tables by properly specifying the primary keys and the foreign keys.
(ii) Enter at least five tuples for each relation.
(iii) Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.
(iv) Find the author of the book which has maximum sales.
(v) Demonstrate how you increase the price of books published by a specific publisher by 10%.
(vi) Generate suitable reports.
(vii) Create suitable front end for querying and displaying the results.

V. Consider the following database for a banking enterprise

BRANCH (BRANCH_NAME: STRING, BRANCH-CITY: STRING, ASSETS: REAL)

ACCOUNT (ACCNO: INT, BRANCH-NAME: STRING, BALANCE: REAL)

DEPOSITOR (CUSTOMER-NAME: STRING, ACCNO: INT)

CUSTOMER (CUSTOMER -NAME: STRING, CUSTOMER -STREET: STRING, CITY: STRING)

LOAN (LOAN-NO: INT, BRANCH-NAME: STRING, AMOUNT REAL)

BORROWER (CUSTOMER-NAME: STRING, LOAN-NO: INT)

(i) Create the above tables by properly specifying the primary keys and the foreign keys
(ii) Enter at least five tuples for each relation
(iii) Find all the customers who have at least two accounts at the *Main* branch.
(iv) Find all the customers who have an account at *all* the branches located in a specific city.
(v) Demonstrate how you delete all account tuples at every branch located in a specific city.
(vi) Generate suitable reports.
(vii) Create suitable front end for querying and displaying the results.

Instructions:
1. The exercises are to be solved in an RDBMS environment like Oracle or DB2.
2. Suitable tuples have to be entered so that queries are executed correctly.
3. Front end may be created using either VB or VAJ or any other similar tool.
4. The student need not create the front end in the examination. The results of the queries may be displayed directly.
5. Relevant queries other than the ones listed along with the exercises may also be asked in the examination.

6. Questions must be asked based on lots.

Table of Contents

I. Consider the **Insurance Database** given below. The primary keys are underlined and the data types are specified. Write the ER Diagram

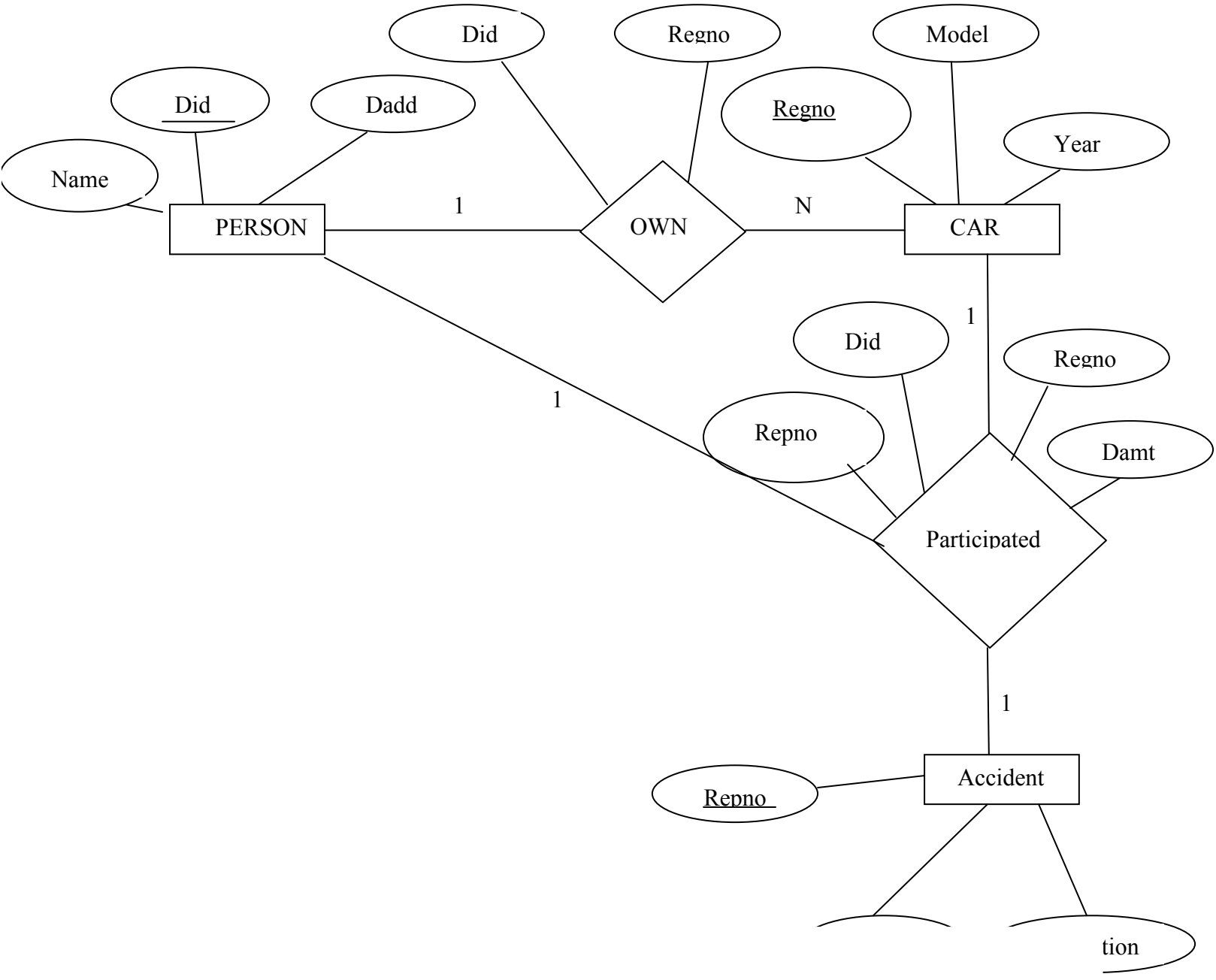
PERSON (DRIVER_ID #: STRING, NAME: STRING, ADDRESS: STRONG)

CAR (REGNO: STRING, MODEL: STRING, YEAR: INT)

ACCIDENT (REPORT_NO: INT, DATE: ADATE, LOCATION: STRING)

OWNS (DRIVER_ID #: STRING. REGNO: STRING)

PARTICIPATED(DRIVER_ID#: STRING, REGNO: STRING, REPORT_NO:INT, DAMAGE_AMT: INT)



1) **Create the above tables by properly specifying the primary keys and the foreign keys.**

```
CREATE TABLE PERSON
        (
        DRIVER_ID# VARCHAR2(10),
        NAME VARCHAR2(20),
         ADDRESS VARCHAR2(15),
         PRIMARY KEY(DRIVER_ID#)
        );

CREATE TABLE CAR
        (
        REGNO VARCHAR2(10),
         MODEL VARCHAR2(10),
         YEAR NUMBER(4),
         PRIMARY KEY(REGNO)
        );

CREATE TABLE ACCIDENT
        (
        REPORT_NO NUMBER(5),
        ADATE DATE,
        LOCATION VARCHAR2(15),
        PRIMARY KEY(REPORT_NO)
        );


CREATE TABLE OWNS
        (
         DRIVER_ID# VARCHAR2(10),
         REGNO VARCHAR2(10),
         PRIMARY KEY(DRIVER_ID#,REGNO),
         FOREIGN KEY(DRIVER_ID#) REFERENCES PERSON(DRIVER_ID#),
         FOREIGN KEY(REGNO) REFERENCES CAR(REGNO)
        );


CREATE TABLE PARTICIPATED
        (
         DRIVER_ID# VARCHAR2(10),
         REGNO VARCHAR2(10),
         REPORT_NO NUMBER(5),
         DAMAGE_AMT NUMBER(7,2),
         PRIMARY KEY(DRIVER_ID#,REGNO,REPORT_NO),
         FOREIGN KEY(DRIVER_ID#,REGNO) REFERENCES OWNS(DRIVER_ID#,REGNO) ON
        DELETE CASCADE,
        FOREIGN KEY(REPORT_NO) REFERENCES ACCIDENT(REPORT_NO) ON
        DELETE CASCADE
        );
```

2) Enter at least five tuples for each relation.

```
INSERT INTO PERSON VALUES
  ('&DRIVER_ID#','&NAME','&ADDRESS');

INSERT INTO CAR VALUES
   ('&REGNO','&MODEL',&YEAR);

INSERT INTO ACCIDENT VALUES
   (&REPORT_NO,'&ADATE','&LOCATION');

INSERT INTO OWNS VALUES
   ('&DRIVER_ID#','&REGNO');

INSERT INTO PARTICIPATED VALUES
   ('&DRIVER_ID#','&REGNO','&REPORT_NO',&DAMAGE_AMT);
```

3) View the records

PERSON

DRIVER_ID#	NAME	ADDRESS
1111	RAMU	K.S.LAYOUT
2222	JOHN	INDIRANAGAR
3333	PRIYA	JAYANAGAR
4444	GOPAL	WHITEFIELD
5555	LATHA	VIJAYANAGAR

CAR

REGNO	MODEL	YEAR
KA04Q2301	MARUTHI-DX	2000
KA05P1000	FORDICON	2000
KA03L1234	ZEN-VXI	1999
KA03L9999	MARUTH-DX	2002
KA01P4020	INDICA-VX	2002

ACCIDENT

REPORT_NO	ADATE	LOCATION
12	01-JUN-02	M G ROAD
200	10-DEC-02	DOUBLEROAD
300	23-JUL-99	M G ROAD
25000	11-JUN-00	RESIDENCY ROAD
26500	16-OCT-01	RICHMOND ROAD

OWNS

DRIVER_ID#	REGNO
1111	KA04Q2301
1111	KA05P1000
2222	KA03L1234
3333	KA03L9999
4444	KA01P4020

PARTICIPATED

DRIVER_ID#	REGNO	REPORT_NO	DAMAGE_AMT
1111	KA04Q2301	12	20000
2222	KA03L1234	200	500
3333	KA03L9999	300	10000
4444	KA01P4020	25000	2375
1111	KA05P1000	26500	70000
2222	KA03L9999	11	10000

4) Demonstrate how you update damage amount for the car with a specific regno in accident with reportno 12 to 25000

```
UPDATE PARTICIPATED
  SET  DAMAGE_AMT=25000
    WHERE REPORT_NO =12 AND REGNO='KA04Q2301';
```

Output

PARTICIPATED

DRIVER_ID#	REGNO	REPORT_NO	DAMAGE_AMT
1111	KA04Q2301	12	25000
2222	KA03L1234	200	500
3333	KA03L9999	300	10000
4444	KA01P4020	25000	2375
1111	KA05P1000	26500	70000
2222	KA03L9999	12	10000

Add a new accident to the database

Insert into accident values (11,'05-MAY-2002','RAJAJINAGAR');

Output

>Select * from accident;

ACCIDENT

REPORT_NO	ADATE	LOCATION
12	01-JUN-02	M G ROAD
200	10-DEC-02	DOUBLEROAD
300	23-JUL-99	M G ROAD
25000	11-JUN-00	RESIDENCY ROAD
26500	16-OCT-01	RICHMOND ROAD
11	05-MAY-02	RAJAJINAGAR

5) Find the total no of people who owned cars that were involved in accidents 2002.

```
SELECT COUNT(O.DRIVER_ID#) FROM OWNS O,ACCIDENT A,PARTICIPATED P
WHERE A.REPORT_NO=P.REPORT_NO AND O.DRIVER_ID#=P.DRIVER_ID# AND A.ADATE
LIKE '__-__-2002';
```

Output

```
COUNT (O.DRIVER_ID#)
--------------------------------
               3
```

6) Find the number of accidents in which cars belonging to a specific model were involved.

```
SELECT COUNT(*),MODEL
  FROM ACCIDENT A, PARTICIPATED P, CAR C
    WHERE A.REPORT_NO=P.REPORT_NO
```

```
AND
 P.REGNO=C.REGNO
  AND
   C.MODEL='MARUTHI-DX';
```

Output

```
COUNT (*)          MODEL
---------------    ------------
      1            MARUTHI-DX
```

II. Consider the following relations for an **Order Processing Database** application in a company. The primary keys are underlined and the data types are specified. Write the ER Diagram

CUSTOMER (CUST #: INT, CNAME: STRING, CITY: STRING)

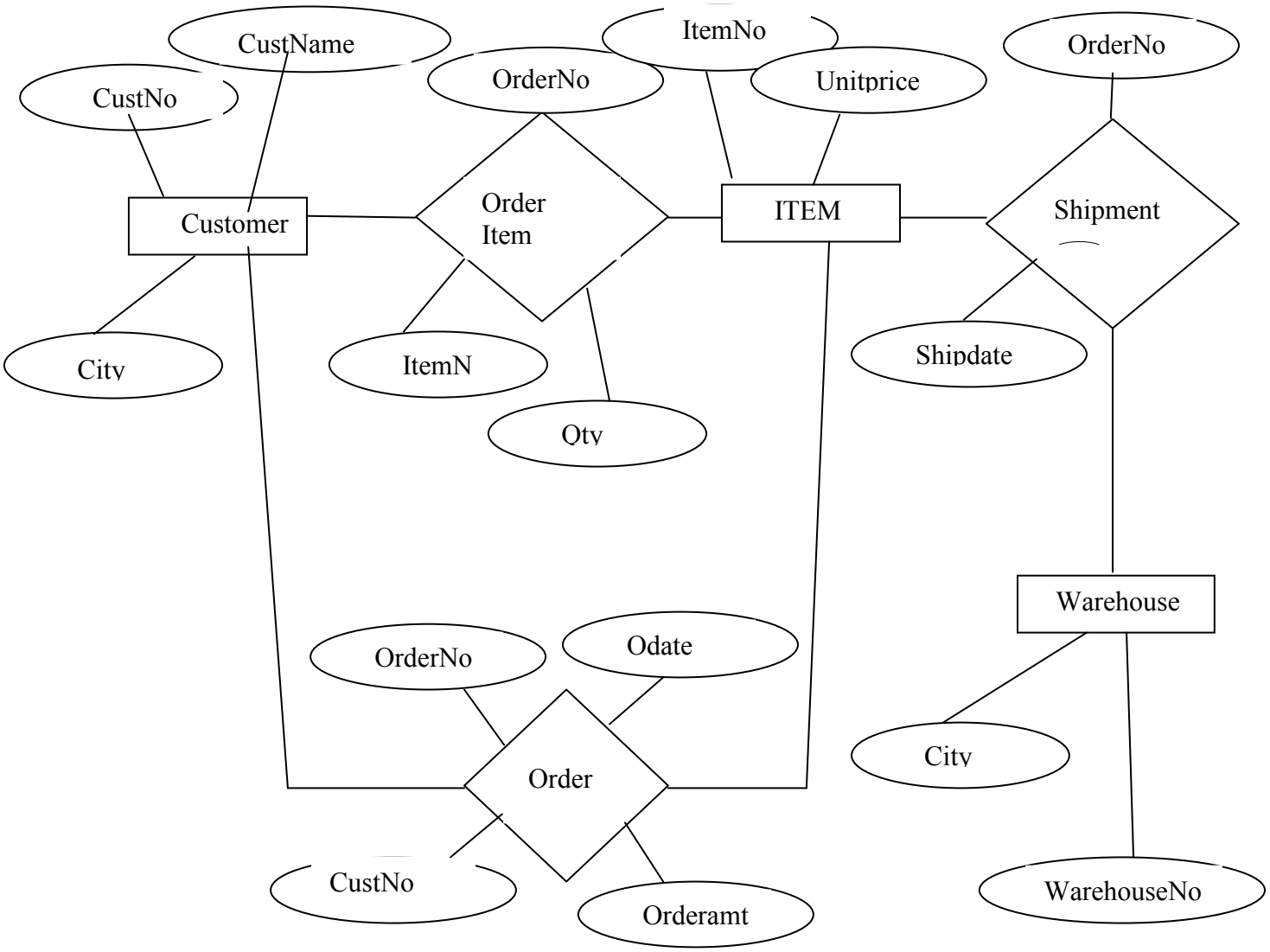
ORDER (ORDER #: INT, ODATE: DATE, CUST #: INT, ORD-AMT: INT)

ITEM (ITEM #: INT, UNIT PRICE: INT)

ORDER - ITEM (ORDER #: INT, ITEM #: INT, QTY: INT)

SHIPMENT (ORDER #: INT, WAREHOUSE#: INT, SHIP-DATE: DATE)

WAREHOUSE (WAREHOUSE #: INT, CITY: STRING)



1) Create the above tables by properly specifying the primary keys and The foreign keys

```
CREATE TABLE CUSTOMER
(
CUST#   NUMBER(5),
CNAME VARCHAR2(15) NOT NULL,
CITY VARCHAR2(15),
PRIMARY KEY(CUST#)
);

CREATE TABLE CUSTORDER
(
 ORDER# NUMBER(5) CONSTRAINT PK1 PRIMARY KEY,
 ODATE DATE,
 CUST# NUMBER(5) REFERENCES CUSTOMER(CUST#),
 ORD_AMT NUMBER(5)
 );

CREATE TABLE ITEM
(
 ITEM  NUMBER(5) CONSTRAINT PK2 PRIMARY KEY,
UNITPRICE NUMBER(9,2) NOT NULL
 );

CREATE TABLE ORDER_ITEM
(
ORDER# NUMBER (5),
ITEM NUMBER (5),
QTY NUMBER(4),
FOREIGN KEY(ORDER#) REFERENCES CUSTORDER(ORDER#),
FOREIGN KEY(ITEM) REFERENCES ITEM(ITEM),
PRIMARY KEY(ORDER#,ITEM)
 );


 CREATE TABLE WAREHOUSE
 (
 WAREHOUSE#  NUMBER(5),
CITY VARCHAR2(15),
 PRIMARY KEY(WAREHOUSE#)
  );

CREATE TABLE SHIPMENT
 (
ORDER# NUMBER(5),
WAREHOUSE# NUMBER(5),
SHIP_DATE DATE,
FOREIGN KEY(ORDER#) REFERENCES CUSTORDER(ORDER#),
FOREIGN KEY(WAREHOUSE#) REFERENCES
                WAREHOUSE(WAREHOUSE#),
```

```
PRIMARY KEY(ORDER#,WAREHOUSE#)
 );
```

2) Insert the records into the relations

```
INSERT INTO CUSTOMER VALUES(&CUSTNO,'&CNAME','&CITY');
INSERT INTO CUSTORDER
        VALUES(&ORDER_NO,'&ODATE',&CUSTNO,&ORD_AMT);
INSERT INTO ITEM VALUES(&ITEM_NO,&UNITPRICE);
INSERT INTO ORDER_ITEM VALUES(&ORDER_NO,&ITEM_NO,&QTY);
INSERT INTO WAREHOUSE VALUES(&WAREHOUSE_NO,'&CITY');
INSERT INTO SHIPMENT
        VALUES(&ORDER_NO,&WAREHOUSE_NO,'&SHIPDATE');
```

3) VIEW THE RELATIONS

CUSTOMER

CUST#	CNAME	CITY
1	ABC	BLORE
2	DEF	KOLAR
3	GHI	BLORE
4	JKL	CHITTOR
5	MNO	MYSORE

ORDER

ORDER#	ODATE	CUST#	ORD_AMT
1	01-JAN-06	2	5000.5
2	26-APR-06	3	2500
3	27-APR-06	3	1000
4	30-APR-06	5	1000
5	25-MAY-06	1	5000

ITEM

ITEM#	UNITPRICE
1	2500
2	5000
3	1000
4	5
5	200

ORDER_ITEM

ORDER#	ITEM#	QTY
1	2	1
1	4	1
2	1	1
3	5	5
4	2	2

WAREHOUSE

WAREHOUSE#	CITY
1	BLORE
2	KOLAR
3	CHITTOR
4	MLORE
5	MYSORE

SHIPMENT

ORDER#	WAREHOUSE#	SHIPDATE
1	1	30-APR-06
2	2	29-APR-06
3	2	24-APR-06
4	5	30-APR-06
5	3	01-JUN-06
6	1	01-JUN-06

3) Producing the listing: custname, order#, avg_order_amt, where the middle column is the total no of orders made by the customer, and the last column is the average order amount for the customer

```
SELECT C.CNAME, COUNT(*),AVG(CO.ORD_AMT)
FROM CUSTOMER C, ORDER O
WHERE
C.CUST#=O.CUST# GROUP BY C.CNAME;
```

OUTPUT

CNAME	COUNT(*)	AVG
GHI	2	1750
MNO	1	10000
ABC	1	5000
DEF	1	5000.5

4) List the order# for orders that where shipped from all the warehouse that the company has in specific city.

```
SELECT  ORDER#
 FROM WAREHOUSE W,  SHIPMENT S
WHERE W.WAREHOUSE#=S.WAREHOUSE# AND CITY='BLORE';
```

OREDER#
1

5) Demonstrate how to delete item 10 from the item table and make that field null in the ORDER-ITEM table

```
SELECT CONSTRAINT_NAME,CONSTRAINT_TYPE
FROM USER_CONSTRAINTS
WHERE TABLE_NAME='ORDER_ITEM';
CONSTRAINT_NAME                C
------------------------------ -
SYS_C002734                    P
SYS_C002735                    R
SYS_C002736                    R
```

```
ALTER TABLE ORDER_ITEM DROP CONSTRAINT SYS_C002736;

ALTER TABLE ORDER_ITEM ADD CONSTRAINT FK1 FOREIGN KEY(ITEM#)
REFERENCES ITEM(ITEM#) ON DELETE SET NULL;
```

DELETE FROM ITEM WHERE ITEM#=5;

SELECT * FROM ITEM

ITEM#	UNITPRICE
1	2500
2	5000
3	1000
4	-5

SELECT * FROM ORDER_ITEM

ORDER#	ITEM#	QTY
1	**2**	**1**
1	**4**	**1**
2	**1**	**1**
3		**5**
4	**2**	**2**

III. Consider the following database of **student enrollment** in courses & books adopted for each course. The primary keys are underlined and the data types are specified. Write the ER Diagram

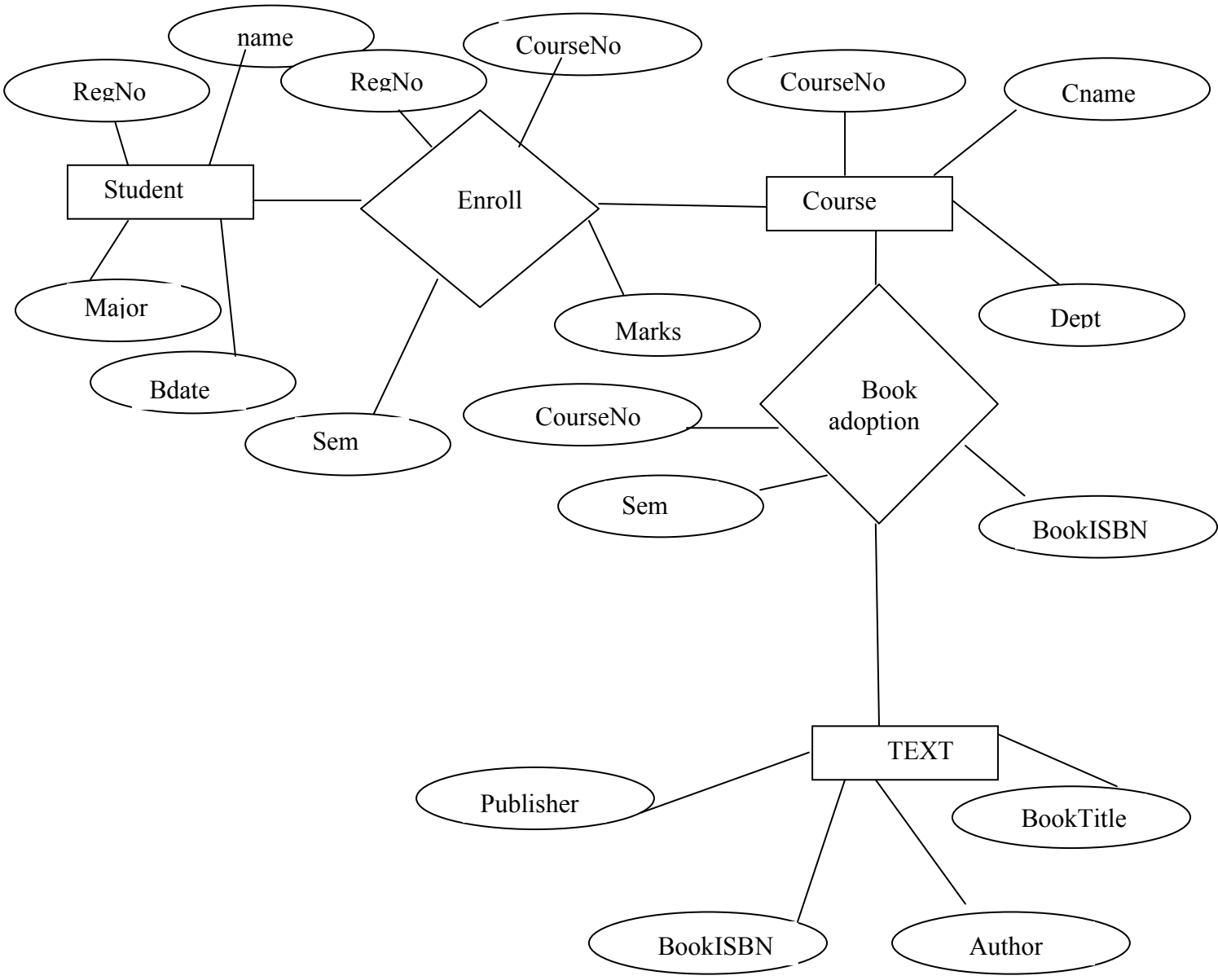
STUDENT (REGNO: STRING, NAME: STRING, MAJOR: STRING, BDATE :DATE)

COURSE (COURSE #: INT, CNAME: STRING, DEPT: STRING)

ENROLL (REGNO: STRING, COURSE#: INT, SEM: INT, MARKS: INT)

BOOK - ADOPTION (COURSE#: INT, SEM: INT, BOOK-ISBN: INT)

TEXT (BOOK-ISBN: INT, BOOK-TITLE: STRING, PUBLISHER:STRING, AUTHOR: STRING)



1) Create the above tables by properly specifying the primary keys and the foreign keys

```
CREATE TABLE STUDENT
(
 REGNO VARCHAR2(10),
 NAME VARCHAR2(14),
 MAJOR VARCHAR2(10),
 BDATE DATE,
 PRIMARY KEY(REGNO)
);

CREATE TABLE COURSE
(
 COURSE# NUMBER(4),
 CNAME VARCHAR2(14),
 DEPT VARCHAR2(10),
 PRIMARY KEY(COURSE#)
);

CREATE TABLE ENROLL
(
 REGNO VARCHAR2(10),
 COURSE# NUMBER(4),
 SEM NUMBER(4),
 MARKS NUMBER(3),
 PRIMARY KEY(REGNO,COURSE#,SEM),
 FOREIGN KEY(REGNO) REFERENCES STUDENT(REGNO),
 FOREIGN KEY(COURSE#) REFERENCES COURSE(COURSE#)
);
CREATE TABLE TEXT
(
 ISBN NUMBER(5),
 BOOK_TITLE VARCHAR2(13) NOT NULL,
 PUBLISHER VARCHAR2(12),
 AUTHOR VARCHAR2(12),
 PRIMARY KEY(ISBN)
);
```

```
CREATE TABLE BOOK_ADOPTION
(
 COURSE# NUMBER(5),
 SEM NUMBER(3),
```

```
 ISBN NUMBER(4),
 PRIMARY KEY(COURSE#,SEM),
 FOREIGN KEY(COURSE#) REFERENCES COURSE(COURSE#),
 FOREIGN KEY(ISBN) REFERENCES TEXT(ISBN)
);
```

2)INSERTION OF RECORDS INTO THE RELATIONS

- INSERT INTO STUDENT VALUES('&REGNO','&NAME','&MAJOR','&BDATE');
- INSERT INTO COURSE VALUES(&COURSE#,'&CNAME','&DEPT');
- INSERT INTO ENROLL VALUES('&REGNO',&COURSE#,&SEM,&MARKS);
- INSERT INTO TEXT VALUES(&ISBN,'&BOOK_TITLE','&PUBLISHER','&AUTHOR');
- INSERT INTO BOOK_ADOPTION VALUES(&COURSE#,&SEM,&ISBN);

VIEW THE RECORDS OF THE RELATIONS

STUDENT

REGNO	NAME	MAJOR	BDATE
1DA05CS045	A	BIOLOGY	25-DEC-84
1DA05CS062	B	CHE	23-JAN-86
1DA05CS015	C	PHYSICS	20-JUN-86
1DA05CS025	D	MA	30-MAR-84
1DA05IS405	E	BIO	06-APR-84

COURSE

COURSE#	CNAME	DEPT
1	MCA	MANAG
2	MBA	MANAG
3	ISE	CS
4	CSE	CS
5	CIV	CIVIL

ENROLL

REGNO	COURSE#	SEM	MARKS
1DA05CS062	2	3	85
1DA05CS015	3	5	57
1DA05CS025	4	8	92
1DA05CS025	4	4	91
1DA05IS405	5	2	85

TEXT

BOOK_ISBN	BOOK_TITLE	PUBLISHER	AUTHOR
111001	DBMS	TATA	NAVATHE
111002	CN	PDR	TENENBAUM
111003	DS	MC-MIILAN	GALVIN
111004	ADA	PERSON	ULLMAN
111005	SE	PRESS	PRESSMAN

BOOK_ADOPTION

COURSE#	SEM	BOOK_ISBN
1	1	111001
2	2	111004
3	4	111003
4	1	111005
5	5	111006
6	4	111003
3	5	111001

3) DEMONSTRATE HOW YOU ADD NEW TEXT BOOK TO THE DATABASE AND MAKE THAT BOOK IS ADOPTED BY SOME DEPARTMENT.

```
INSERT INTO TEXT VALUES(111006,'AMP','MHP,'BERY);
```

```
INSERT INTO BOOK_ADOPTION VALUES(5,5,111006);
```

4) PRODUCE A LIST OF TEXTBOOKS(INCLUDE COURSE,BOOK_ISBN,BOOK_TITLE) IN THE ALPHABETIC ORDER FOR COURSES OFFERED BY THE 'CSE' DEPT THAT USE MORE THAN TWO BOOKS.

```
SELECT C.COURSE#, B.ISBN,BOOK_TITLE
    FROM COURSE C,BOOK_ADOPTION B,TEXT T
      WHERE C.COURSE#=B.COURSE# AND B.ISBN=T.ISBN  AND
        C.COURSE# IN   (SELECT C1.COURSE# FROM COURSE
            C1,BOOK_ADOPTION B1
                  WHERE
              C1.COURSE#=B1.COURSE# AND DEPT='CSE'
                GROUP BY C1.COURSE#  HAVING COUNT(*)>2)
               ORDER BY CNAME;
```

OUTPUT

COURSE#	BOOK_ISBN	BOOK_TITLE
3	111003	CS
3	111001	DBMS
4	111005	SE
4	111003	OS
4	111006	AMP

5) LIST ANY DEPARTMENT THAT HAS ALL ITS ADOPTED BOOKS PUBLISHED BY A SPECIFIC PUBLISHER

```
SELECT DISTINCT C.DEPT
 FROM COURSE C, BOOK_ADOPT B, TEXT T
  WHERE T.BOOK_ISBN=B.BOOK_ISBN
   AND B.COURSE#=C.COURSE#
                AND T.PUBLISHER='TATA';
```

OUTPUT

DEPT
CS

IV. The following tables are maintained by a **book dealer**. The primary keys are underlined and the data types are specified. Write the ER Diagram

AUTHOR (AUTHOR-ID: INT, NAME: STRING, CITY: STRING, COUNTRY: STRING)

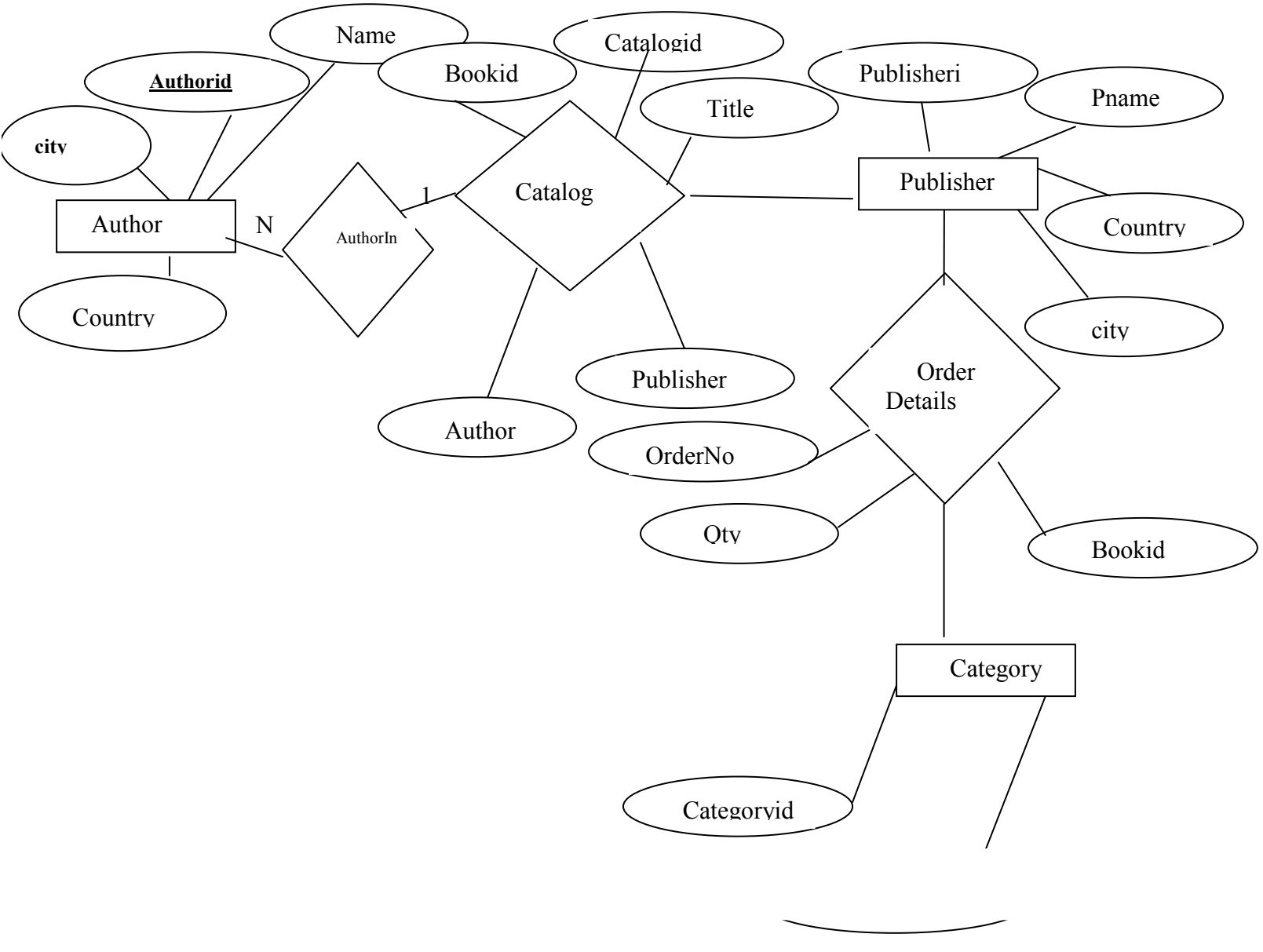
PUBLISHER (PUBLISHER-ID: INT, NAME: STRING, CITY: STRING, COUNTRY: STRING)

CATALOG (BOOK-ID: INT, TITLE: STRING, AUTHOR-ID: INT, PUBLISHER-ID: INT, CATEGORY-ID: INT,

YEAR: INT, PRICE: INT)

CATEGORY (CATEGORY-ID: INT, DESCRIPTION: STRING)

ORDER-DETAILS (ORDER-NO: INT, BOOK-ID: INT, QUANTITY: INT)



1) **Create the above tables by properly specifying the primary keys and the foreign keys**

```
CREATE TABLE AUTHOR
(
 AUTHORID NUMBER(5),
 NAME VARCHAR(15),
 CITY VARCHAR2(15),
 ACOUNTRY VARCHAR(15),
 PRIMARY KEY(AUTHORID)
);

CREATE TABLE PUBLISHER
(
 PUBLISHERID NUMBER(5),
NAMEVARCHAR(15),
 CITY VARCHAR2(15),
 COUNTRY VARCHAR(15),
 PRIMARY KEY(PUBLISHERID)
 );

CREATE TABLE CATEGORY
(
 CATEGORYID NUMBER(5),
 DESCRIPTION VARCHAR(15),
 PRIMARY KEY(CATEGORYID)
 );

 CREATE TABLE CATALOG
 (
  BOOKID NUMBER(5),
 TITLE VARCHAR2(15),
 AUTHORID NUMBER(5),
 PUBLISHERID NUMBER(5),
 CATEGORYID NUMBER(5),
 YEAR NUMBER(5),
 PRICE NUMBER(10,3),
 PRIMARY KEY(BOOKID),
 FOREIGN KEY(AUTHORID) REFERENCES AUTHOR(AUTHORID),
 FOREIGN KEY(PUBLISHERID) REFERENCES PUBLISHER(PUBLISHERID),
 FOREIGN KEY(CATEGORYID) REFERENCES CATEGORY(CATEGORYID)
 );

 CREATE TABLE ORDER_DETAILS
 (
  ORDERNO NUMBER(5),
  BOOKID NUMBER(5),
 QUANTITY NUMBER(5),
 PRIMARY KEY(ORDERNO,BOOKID),
 FOREIGN KEY(BOOKID) REFERENCES CATALOG(BOOKID)
```

);

2) INSERT THE RECORDS INTO THE RELATIONS

```
INSERT INTO AUTHOR
        VALUES(&AUTHORID,'&NAME','&CITY','&ACOUNTRY');
INSERT INTO PUBLISHER
        VALUES(&PUBLISHERID,'&NAME','&CITY','&COUNTRY');
INSERT INTO CATEGORY VALUES(&CATEGORYID,'&DESCRIPTION');
INSERT INTO CATALOG VALUES
       (&BOOKID,'&TITLE',&AUTHORID,&PUBLISHERID,&CATEGORYID,&YEAR,
        &PRICE);
INSERT INTO ORDER_DETAILS VALUES(&ORDERNO,&BOOKID,&QUANTITY);
```

3) VIEW THE RELATIONS

AUTHOR

AUTHORID	NAME	CITY	COUNTRY
101	ABC	DELHI	INDIA
102	TONY	HAYHOOD	USA
103	GHI	PATNA	INDIA
104	JKL	BELM	SRILANKA
105	MND	BANGALORE	INDIA

PUBLISHER

PUBLISHERID	NAME	CITY	COUNTRY
1001	pbp	blore	INDIA
1002	palk	slaugh	england
1003	press	tata	INDIA
1004	rathe	angakara	srilanka
1005	pbp	blore	india

CATEGORY

CATEGORYID	DESCRIPTIONM
10001	cs
10002	med
10003	bio
10004	meteor
10005	mech

CATALOG

BOOKID	TITLE	AUTHORID	PUBLISHERID	CATEGORYID	YEAR	PRICE
1000001	dbms	101	1001	10001	1998	235
1000002	or	101	1002	10003	1997	255
1000003	cn	102	1003	10002	2001	352
1000004	se	102	1003	10001	2002	523
1000005	ada	103	1004	10004	2003	124

ORDER_DETAILS

ORDER-NO	BOOK-ID	QUANTITY
1	1000001	12
1	1000002	2
2	1000002	15
3	1000003	23
4	1000003	14
5	1000005	7

4) GIVE THE DETAILS OF THE AUTHORS WHO HAVE TWO OR MORE BOOKS IN THE CATALOG AND THE PRICE OF THE BOOKS IS GREATER THAN THE AVERAGE PRICE OF THE BOOKS IN THE CATALOG & THE YEAR OF PUBLICATION IS AFTER 2000.

```
SELECT * FROM AUTHOR A WHERE A.AUTHORID IN
        (SELECT C.AUTHORID FROM CATALOG C
                WHERE YEAR>2000 AND
                    C.PRICE > (SELECT AVG (PRICE) FROM CATALOG)
                                GROUP BY C.AUTHORID HAVING COUNT (AUTHOR-ID)>=2);
```

Output

AUTHOR-ID	NAME	CITY	COUNTRY
102	DEF	JAMES	INDIA

5) FIND THE AUTHOR OF THE BOOK WHICH HAS MAXIMUN SALES

```
SELECT NAME FROM AUTHOR WHERE AUTHORID IN
  (
```

SELECT AUTHORID FROM CATALOG , ORDER_DETAILS O WHERE O.BOOK-ID=CATALOG.BOOK-ID AND QUANTITY=9SELECT MAX(QUANTITY) FROM OREDER-DETAILS));

NAME

DEF

6) DEMONSTRATE HOW YOU INCREASE THE PRICE OF BOOK PUBLISHED BY A SPECIFIC PUBLISHER BY 10%.

```
UPDATE CATALOG SET PRICE=PRICE+PRICE*0.10
  WHERE PUBLISHERID IN (SELECT P.PUBLISHERID
                                    FROM PUBLISHER P
                                        WHERE P.NAME=''PBP');
```

Output

SQL>SELECT * FROM CATALOG;

BOOKID	TITLE	AUTHORID	PUBLISHERID	CATEGORYID	YEAR	PRICE
1000001	dbms	101	1001	10001	1998	258
1000002	Or	101	1002	10003	1997	255
1000003	Cn	102	1003	10002	2001	352
1000004	Se	102	1003	10001	2002	523
1000005	ada	103	1004	10004	2003	124

V. Consider the following database for a **banking enterprise**. The primary keys are underlined and the data types are specified. Write the ER Diagram

BRANCH (BRANCH_NAME: STRING, BRANCH-CITY: STRING, ASSETS: REAL)

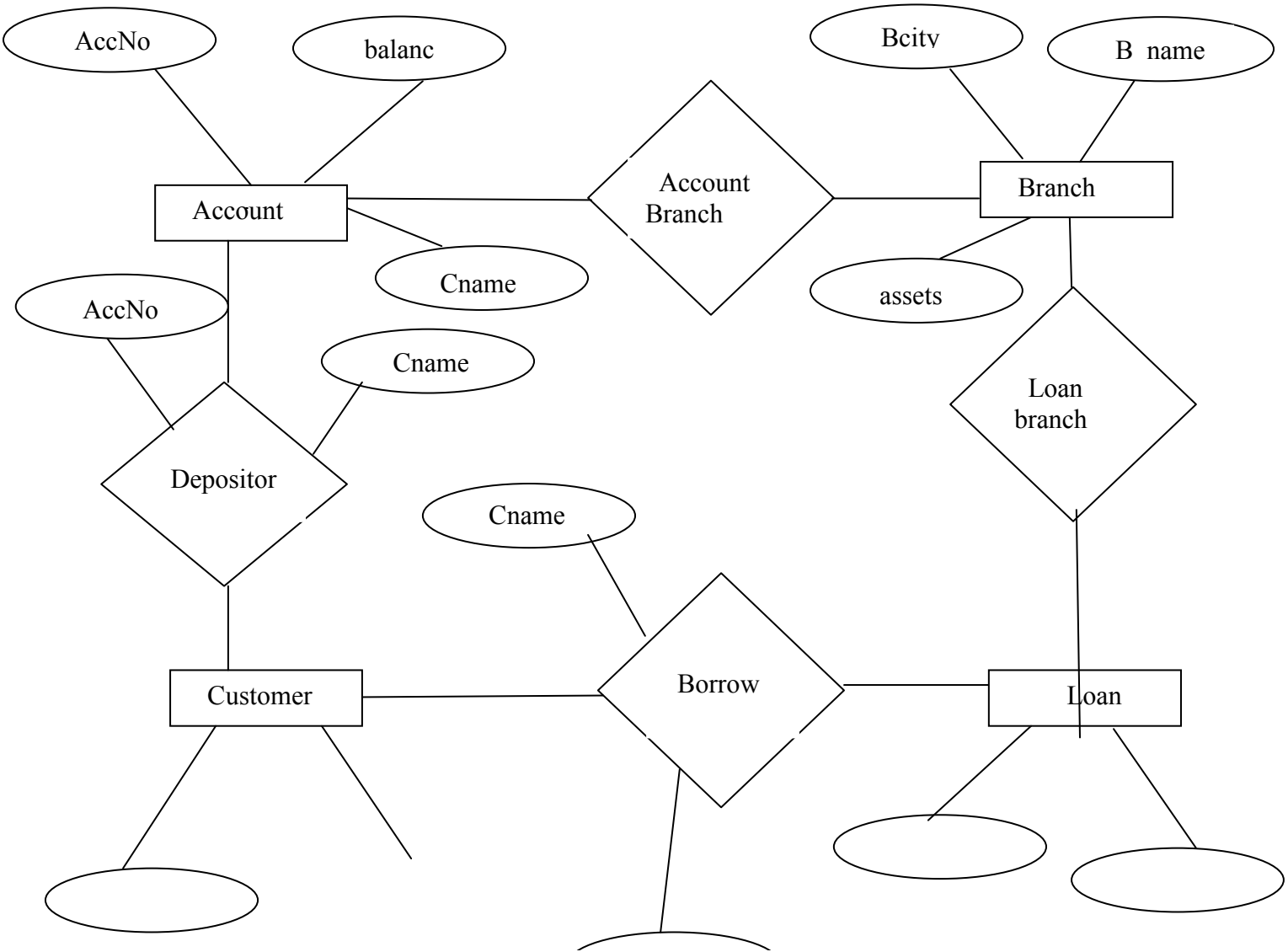
ACCOUNT (ACCNO: INT, BRANCH-NAME: STRING, BALANCE: REAL)

DEPOSITOR (CUSTOMER-NAME: STRING, ACCNO: INT)

CUSTOMER (CUSTOMER -NAME: STRING, CUSTOMER -STREET: STRING, CITY: STRING)

LOAN (LOAN-NO: INT, BRANCH-NAME: STRING, AMOUNT REAL)

BORROWER (CUSTOMER-NAME: STRING, LOAN-NO: INT)



1) CREATION OF TABLES WITH NECESSARY CONSTRAINTS

```
CREATE TABLE BRANCH
(
 BRANCH_NAME VARCHAR2(15),
 BRANCH_CITY VARCHAR2(15),
 ASSETS NUMBER(10,3),
 PRIMARY KEY(BRANCH_NAME),
 CHECK(ASSETS>0)
);

CREATE TABLE CUSTOMER
(
 CUSTOMER_NAME VARCHAR2(15),
 CUSTOMER_ADDRESS VARCHAR2(15),
 CUSTOMER_CITY VARCHAR2(15),
 PRIMARY KEY(CUSTOMER_NAME)
 );

CREATE TABLE ACCOUNT
(
 ACCNO VARCHAR2(15),
 BRANCH_NAME VARCHAR2(15),
 BALANCE NUMBER(11,4),
 PRIMARY KEY(ACCNO),
 FOREIGN KEY(BRANCH_NAME) REFERENCES
          BRANCH(BRANCH_NAME) ON DELETE CASCADE
);

CREATE TABLE DEPOSITOR
(
 CUSTOMER_NAME VARCHAR2(15),
 ACCNO VARCHAR2(15),
 PRIMARY KEY(CUSTOMER_NAME,ACCNO),
 FOREIGN KEY(CUSTOMER_NAME) REFERENCES
        CUSTOMER(CUSTOMER_NAME)ON DELETE CASCADE,
FOREIGN KEY(ACCNO) REFERENCES
        ACCOUNT(ACCNO) ON DELETE CASCADE);
 CREATE TABLE LOAN
```

```
(
 LOAN-NO NUMBER(6),
 BRANCH_NAME VARCHAR2(15),
 AMOUNT NUMBER(10,4),
 PRIMARY KEY(LOAN-NO),
 FOREIGN KEY(BRANCH_NAME) REFERENCES
        BRANCH(BRANCH_NAME)ON DELETE CASCADE
);

CREATE TABLE BORROWER
(
 CUSTOMER_NAME VARCHAR2(15),
 LOAN-NO NUMBER(5),
 PRIMARY KEY(CUSTOMER_NAME,LOAN-NO),
 FOREIGN KEY(CUSTOMER_NAME) REFERENCES
        CUSTOMER(CUSTOMER_NAME)ON DELETE CASCADE,
FOREIGN KEY(LOAN-NO) REFERENCES
       LOAN(LOAN-NO)ON DELETE CASCADE
);
```

2)INSERTION OF TUPLES INTO THE RELATION

- INSERT INTO BRANCH VALUES('&BRANCH_NAME','&BRANCH_CITY',&ASSETS);
- INSERT INTO CUSTOMER VALUES('&CUSTOMER_NAME','&CUSTOMER_ADDRESS', '&CUSTOMER_CITY');
- INSERT INTO ACCOUNT VALUES('&ACCNO','&BRANCH_NAME',&BALANCE);
- INSERT INTO DEPOSITOR VALUES('&CUSTOMER_NAME','&ACCNO');
- INSERT INTO LOAN VALUES(&LOAN-NO,'&BRANCH_NAME',&AMOUNT);
- INSERT INTO BORROWER VALUES('&CUSTOMER_NAME',&LOAN-NO);

3)VIEW THE RECORDS

BRANCH

BRANCH-NAME	BRANCHCITY	ASSET
BGUDI	BLORE	100000
JAYNAGAR	BLORE	200000
ANNA	CHENNAI	300000
KAROL	DELHI	400000
ANDHRI	MUMNBAI	500000

CUSTOMER

CUSTOMERNAME	CUSTOMERSTREET	CUSTOMERCITY
AJAY	M G ROAD	BLORE

SUMUKH	BRIGADE ROAD	BLORE
SANTA	MOUNT ROAD	CHENNAI
AMIT	RAJNATH	DELHI
RAJATH	A S STREET	MUMBAI

ACCOUNT

ACCNO	BRANCH-NAME	BALANCE
1	BGUDI	1000
2	JAYNAGAR	5000
3	ANNA	3000
4	KAROL	4000
5	ANDHRI	6000

DEPOSITOR

CUSTOMER-NAME	ACCNO
AJAY	1
SUMUKH	2
SANTA	3
AMIT	4
RAJUTH	5
AJAY	7

LOAN

LOANNO	BRANCHNAME	AMOUNT
10	BGUDI	3000
20	JAYNAGAR	2000
30	ANNA	25000
40	KAROL	4000
50	ANDHRI	3000

BORROWER

LOANNO

CUSTOMERNAME	
AJAY	10
SUMUKH	20
SANTA	30
AMIT	40
RAJATH	50

4) FIND ALL CUSTOMERS WHO HAVE ACCOUNT AT ALL THE BRANCHES LOCATED IN A SPECIFIC CITY

```
>SELECT D.CUSTOMERNAME
FROM DEPOSITOR D, ACCOUNT A, BRANCH B
WHERE D.ACCNO=A.ACCNO AND
        A.BRANCHNAME=B.BRANCHNAME AND
        B.BRANCHCITY='CHENNAI'
GROUP BY D.CUSTOMERNAME
HAVING COUNT(A.BRANCHNAME)=(SELECT COUNT(X.BRANCHNAME)
                            FROM BRANCH X
                            WHERE X.BRANCHCITY='CHENNAI');
```

Output

```
CUSTOMERNAME
------------------------
AJAY
AMIT
RAJANT
SANTA
SUMUKH
```

5) FIND ALL CUSTOMERS WHO HAVE AT LEAST TWO ACCOUNTS AT THE MAIN BRANCH

```
>SELECT D.CUSTOMERNAME
FROM DEPOSITOR D, ACCOUNT A, BRANCH B
WHERE D.ACCNO=A.ACCNO AND
        A.BRANCHNAME=B.BRANCHNAME AND
        B.BRANCHCITY='CHENNAI'
GROUPBY C.CUSTOMERNAME
HAVING COUNT (*)>=2;
```

Output

```
CUSTOMERNAME
-------------------------
```

AJAY

6) DEMONSTRATE HOW YOU DELETE ALL THE TUPLES AT EVERY BRANCH LOCATED IN A PARTICULAR CITY

```
DELETE FROM ACCOUNT
WHERE BRANCHNAME IN ( SELECT B.BRANCHNAME
                      FROM BRANCH B
                      WHERE B.BRANCHCITY='BLORE'
                    );
```

Output

```
SQL> /
3 ROWS DELETED

SQL>SELECT * FROM ACCOUNT;
```

ACCNO	BRANCH-NAME	BALANCE
3	ANNA	3000
4	KAROL	4000
5	ANDHRI	6000

EXTRA QUERIES

1) SUPPOSE THAT ANNUAL INTEREST PAYMENTS ARE BEING MADE AND ALL BRANCHES ARE TO BE INCREASED BY 3%.

2) IF INTEREST IS TO BE PAID ONLY TO ACCOUNTS WITH A BALANCE OF Rs.1000 OR MORE.

3) FIND ALL LOAN NUMBERS FOR LOANS MADE AT KUMBALAGODU BRANCH WITH LOAN AMOUNT ABOVE Rs.5000.

4) FIND THE LOAN NUMBER OF THOSE LOANS WITH LOAN AMOUNTS BETWEEN 10000 AND 40000.

5) FIND ALL THE CUSTORES HAVING A LOAN, AN ACCOUNT OR BOTH AT THE BANK.

6) FIND ALL THE CUSTORES WHO HAVE AN ACCOUNT AND A LOAN AT THE BANK.

7) FIND THE AVERAGE ACCOUNT BALANCE AT EACH BRANCH.

8) FIND THE NUMBER OF DEPOSITORS FOR EACH BRANCH.

9) FIND THE BRANCHES WHERE THE AVERAGE ACCOUNT BALANCE IS MORE THAN Rs.5000.

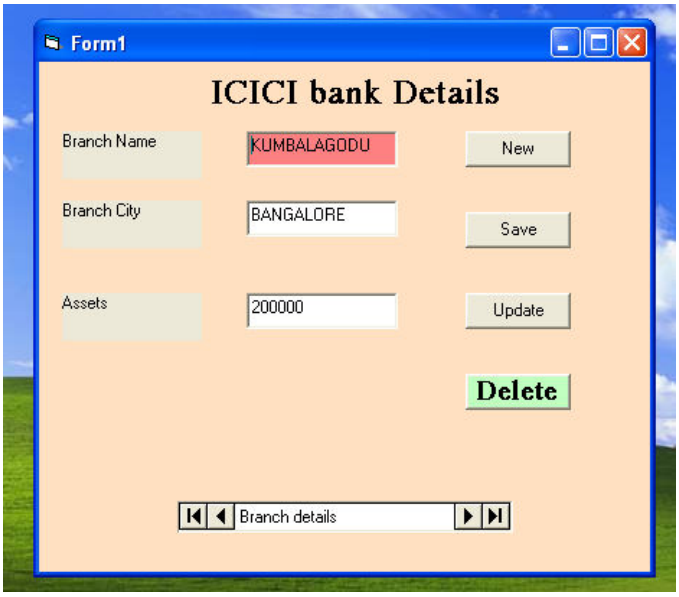
10) FIND THE NAMES OF ALL BRANCHES THAT HAVE AN ASSET VALUE GREATER THAN THAT OF EACH BRANCH IN BANGALORE.

11) FIND THE NAMES OF ALL THE BRANCHES OF LOAN RELATION

Creation of suitable Front End using Visual Basic for querying and displaying the results

STEP 1:

Create the necessary FORM using label, textbox & button controls.

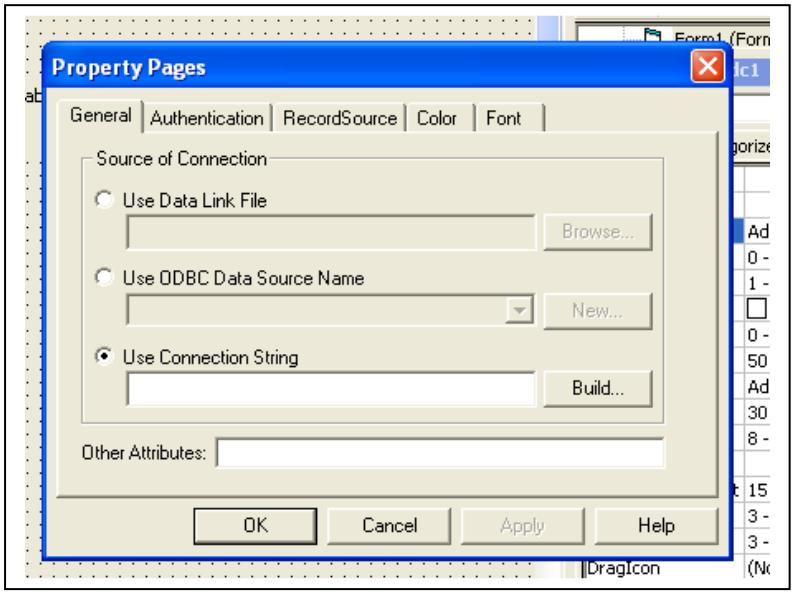


STEP 2:

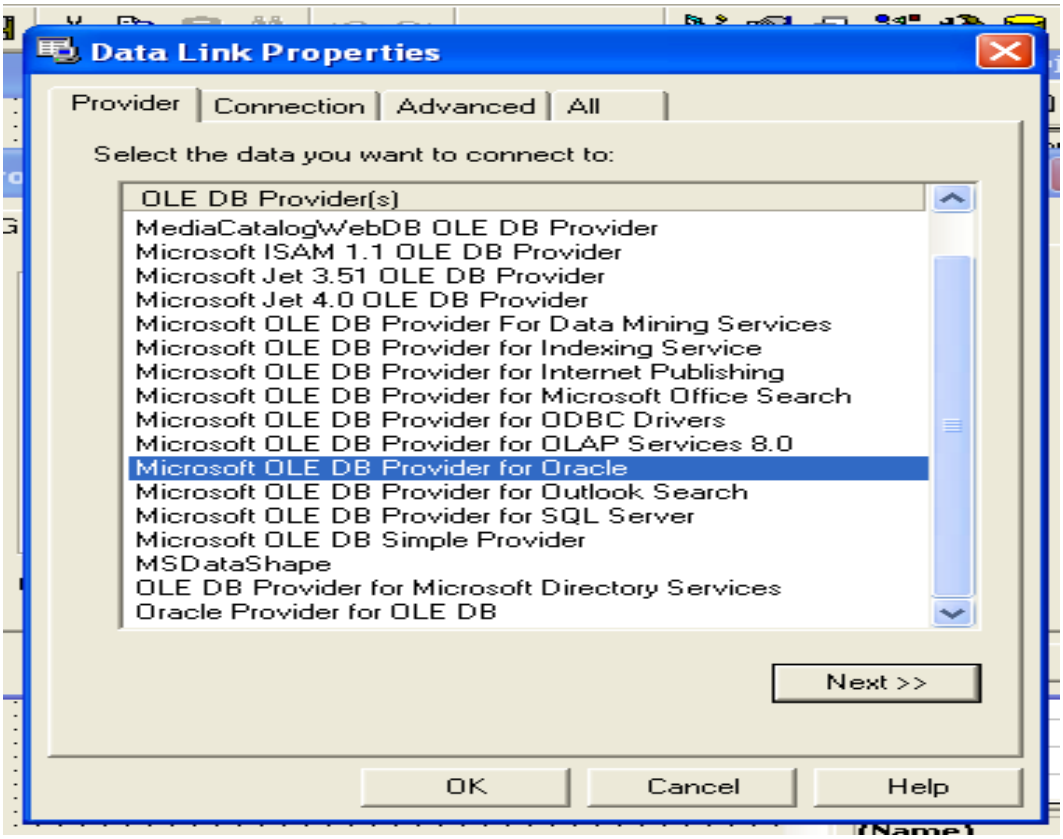
Establish connectivity between User interface screens & Database

1. Select components from the project menu
2. Select **MS ADO Data Control 6.0 (OLE DB)** & click OK

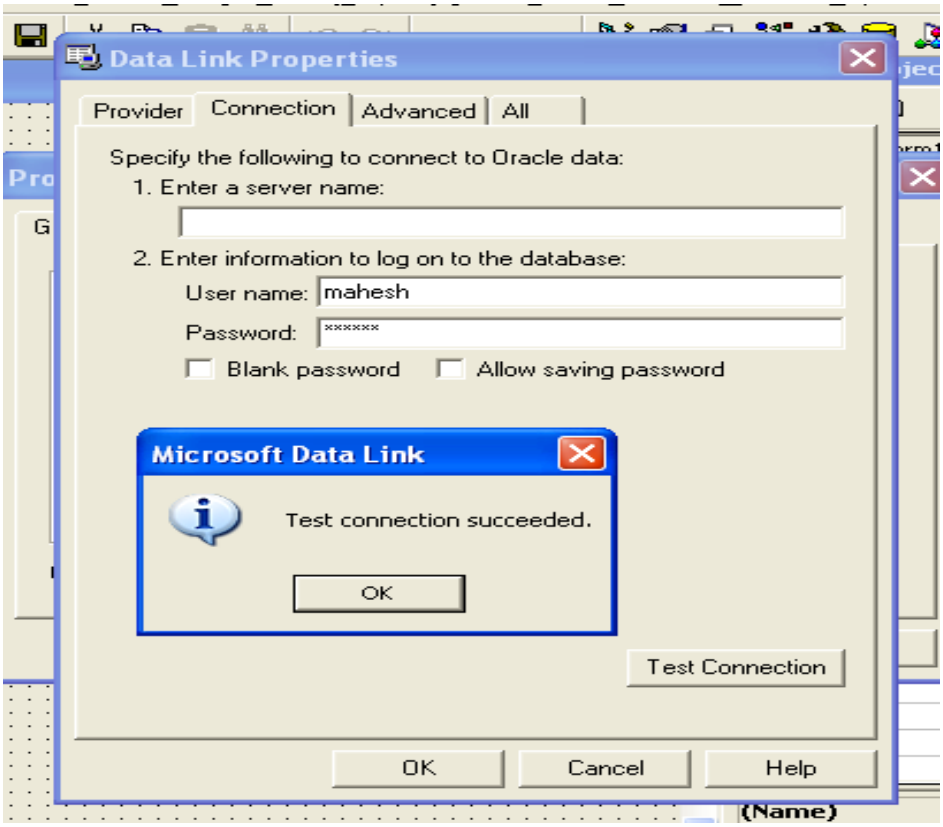
3. Add the ADODC from the ToolBox to the form & name the control **adcBranch**
4. Set ConnectString, CommandType & Record Source properties of the ADO Data Control as follows
5. Right click on adcBranch data control & select ADODC Properties

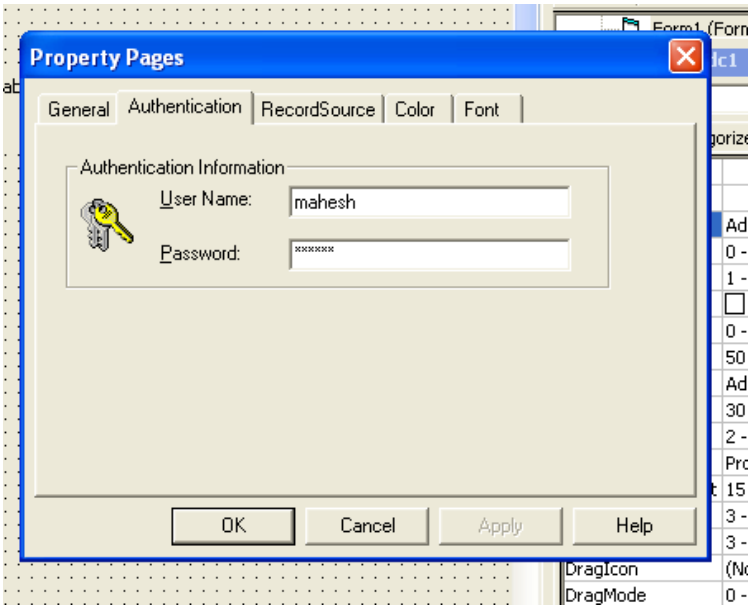


6. To set the connection [property click on BUILD button & you get a Data Link Dialog box as follows

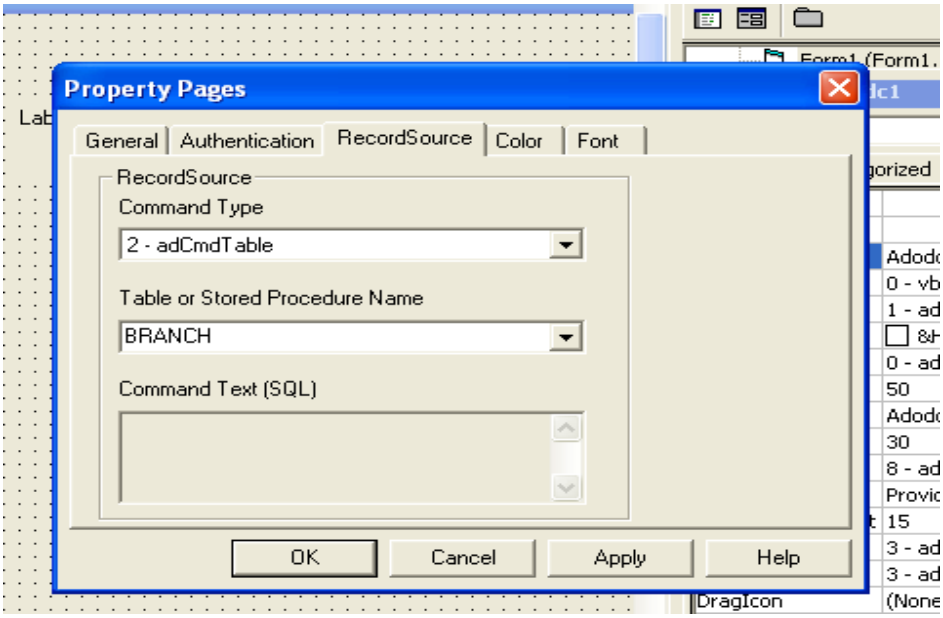


6. Select MS OLE DB Provider for Oracle & Click on NEXT button
7. Enter User name & Password & Test connection
8. Click OK. Click Authentication TAB & enter User Name & password & Click on APPLY button.





9. Click on **RecordSource TAB &** select **2 - adCmdTable.** Wait till the tables are listed in the Table combo Box



10. Select the **BRANCH** table & click on APPLY

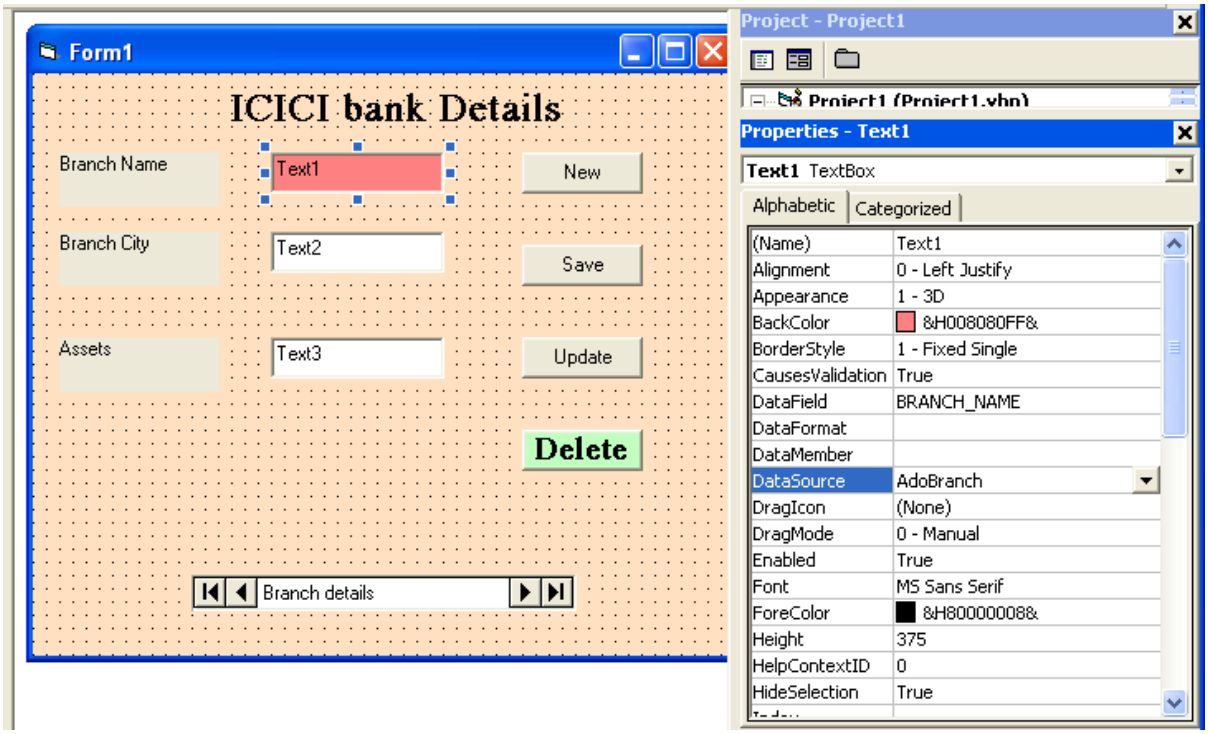
11. Click on OK button to complete the settings

STEP 3:

Binding the FORM Controls to the Table Fields

1. Select the **txtBranchName** control, right click & select properties.

2. In the properties windows set **DataSource & DataField** (In that order) Select **adcBranch in DataSource & BRANCH_NAME in DataField** as shown below.



STEP 4: Add Events to BUTTON controls

1. Double click on the **New button** & add the following code

```
Private Sub New_Click()
        adcBranch.Recordset.AddNew
End Sub
```

2. Double click on the **Save button** & add the following code

```
Private Sub Save_Click()
        adcBranch.Recordset.Save
        MsgBox "Record Added Successfully"
End Sub
```

3. Double click on the **Update button** & add the following code

```
Private Sub Update_Click()
        adcBranch.Recordset.Update
        adcBranch.Recordset.Save
        MsgBox "Record Updated Successfully"
End Sub
```

4. Double click on the **Delete button** & add the following code

```
Private Sub Delete_Click()
   adcBranch.Recordset.Delete
   adcBranch.Recordset.Save
   MsgBox "Record Deleted..."
   adcBranch.Recordset.MoveNext
End Sub
```

STEP 5: Execute the program

1. Select Run menu
2. Click on start sub menu

VIVA QUESTIONS

1). what is DBMS?

1) What are the advantages of DBMS?
2) Differentiate between DBMS and RDBMS.
3) Explain the three schema architecture.
4) What is data independence?
5) Explain different data models.
6) Explain ER model.
7) What is an entity?
8) What is a weak entity?
9) What is total dependency?
10) Explain the different types of attributes.
11) What is Hierarchical model?
12) What is Network model?
13) What is a derived attribute? Give an example.
14) Explain different mapping constraints.
15) Define primary key
16) Define candidate key
17) Define Unique key
18) Define super key
19) What is Referential Integrity?
20) What is a foreign key?
21) Explain DDL commands
22) Explain DML commands
23) Explain DCL commands
24) Explain TCL commands
25) What is normalization?
26) Explain 1st Normal form.
27) Explain 2nd Normal form
28) Explain 3rd Normal form
29) Explain Boyce Codd Normal form
30) What is Functional dependency? Explain.
31) Give an example for group by clause.
32) Give examples for where clause.

33) Give examples for order by clause.
34) What is a transaction?
35) Explain ACID properties.
36) What is 2-phase locking?
37) What is Distributed DBMS?
38) What is Oracle?
39) Give examples for database servers.
40) What do you mean by front end? Give examples.
41) What is ODBC?
42) How do you connect Oracle to Visual Basic?
43) What are the different types of data types used in Oracle 8i?
44) What is the difference between CHAR and VARCHAR data type?
45) What is the difference between DROP and TRUNCATE commands?
46) What is the use of ROLLBACK command?
47) Explain the various uses of ALTER command.
48) Is it possible to increase the size of the data type that is set? Explain.
49) How do you add/drop a column?