# Assignment 1a

**SUPPLIER (SID, SNAME, CITY)**

**PART (PID, PNAME, COLOR)**

**CATALOG (SID, PID, COST)**

1. Create the above tables with necessary constraints.

CREATE TABLE SUPPLIER (

SID INT PRIMARY KEY,

SNAME VARCHAR2(15),

CITY VARCHAR2(10)

);

CREATE TABLE PART (

PID INT PRIMARY KEY,

PNAME VARCHAR2(15),

COLOR VARCHAR2(10)

);

CREATE TABLE CATALOG (

SID INT,

PID INT,

COST INT,

FOREIGN KEY(SID) REFERENCES SUPPLIER(SID),

FOREIGN KEY(PID) REFERENCES PART(PID)

);

1. Display details of suppliers who supply red colored parts.

SELECT \* FROM SUPPLIER

WHERE SID IN (SELECT SID FROM CATALOG

WHERE PID IN (SELECT PID FROM PART WHERE COLOR='RED'));

1. Display total number of suppliers for each part.

SELECT PID, COUNT(SID) AS SP\_NO FROM CATALOG

GROUP BY PID ORDER BY PID;

1. Display suppliers based in Mumbai and whose names have ‘j’ as the third character.

SELECT \* FROM SUPPLIER

WHERE CITY='MUMBAI' AND SNAME LIKE '\_\_J%';

1. Create a view ‘catview’ which stores details about suppliers and parts where the cost is more than 100.

CREATE VIEW CATVIEW AS

(SELECT SID,PID FROM CATALOG WHERE COST>100);

1. Add a column SCONTACT to the table SUPPLIER which always has ‘0’(zero) as the first characters.

ALTER TABLE SUPPLIER

ADD SCONTACT VARCHAR2(11) CONSTRAINT SCONTACT\_CHECK CHECK (SCONTACT LIKE '0%');

1. Create a composite index on table SUPPLIER for the columns SNAME and CITY.

CREATE INDEX idx\_supplier\_name\_city ON SUPPLIER (SNAME, CITY);

1. Display all those suppliers who supply more than three parts.

SELECT \* FROM SUPPLIER

WHERE SID IN (SELECT SID FROM CATALOG GROUP BY SID HAVING COUNT(PID)>3);

1. Display suppliers who are not supplying any part.

SELECT \* FROM SUPPLIER S WHERE NOT EXISTS(SELECT 1 FROM CATALOG WHERE S.SID=CATALOG.SID) ORDER BY SID;

1. Display the average cost of all green parts.

SELECT AVG(COST) GREEN\_COST FROM CATALOG WHERE PID IN (SELECT PID FROM PART WHERE COLOR='GREEN');

**CATEGORY (CAT\_NO, CAT\_TYPE, EXPIRY\_DT)**

**ITEM (ITEMNO, IDESC, UNIT\_OF\_MEASURE, QTY\_ON\_HAND, WEIGHT, CAT\_NO)**

1. Create the above tables with necessary constraints.

CREATE TABLE CATEGORY(

CAT\_NO INT PRIMARY KEY,

CAT\_TYPE VARCHAR2(15) NOT NULL,

EXPIRY\_DT DATE

);

CREATE TABLE ITEM (

ITEM\_NO INT PRIMARY KEY,

IDESC VARCHAR2(20),

UNIT\_OF\_MEASURE VARCHAR2(7),

QTY\_ON\_HAND INT,

WEIGHT NUMBER(6,2),

CAT\_NO INT,

FOREIGN KEY(CAT\_NO) REFERENCES CATEGORY(CAT\_NO)

);

1. Add a constraint to the ITEM table which will allow the QTY\_ON\_HAND to have values between 100 and 1000 only.

ALTER TABLE ITEM

ADD CONSTRAINT QTY\_CHECK CHECK(QTY\_ON\_HAND BETWEEN 100 AND 1000);

1. Display how many items are there in each category.

SELECT COUNT(\*), CAT\_NO FROM CATEGORY C LEFT JOIN ITEM I ON C.CAT\_NO=I.CAT\_NO ORDER BY CAT\_NO;

OR

SELECT COUNT(\*),CAT\_NO FROM ITEM GROUP BY CAT\_NO ORDER BY CAT\_NO;

1. Display the items in descending order of their weights.

SELECT \* FROM ITEM ORDER BY WEIGHTS DESC;

1. Display categories which expire in the month of December.

SELECT \* FROM CATEGORY WHERE TO\_CHAR(EXPIRY\_DT, ’MM’) =12;

1. Display items which have the same weight as that of ITEMNO 116.

SELECT \* FROM ITEM

WHERE WEIGHT = (SELECT WEIGHT FROM ITEM WHERE ITEM\_NO=116);

1. Display items which do not have the alphabet ‘A’ or ‘T’ in their IDESC.  
   SELECT \* FROM ITEM WHERE IDESC NOT LIKE ‘%A%’ AND IDESC NOT LIKE ‘%T%’;
2. Increase the size of the column IDESC by making it a variable length of 25 characters.

ALTER TABLE ITEM

MODIFY COLUMN IDESC VARCHAR2(25);

1. Display the item details and the category details of items which have a UNIT\_OF \_MEASURE as Kilogram, Meters or Litres.

SELECT \* FROM CATEGORY C JOIN ITEM I ON C.CAT\_NO = I.CAT\_NO WHERE I.UNIT\_OF\_MEASURE IN (‘Kilogram’,’Meters’,’LItres’)

# Assignment 1b

**TEAM\_MASTER (TEAM\_ID, TEAM\_NAME)**

**PLAYER (TEAM\_ID, PID, PNAME, PBIRTHDATE)**

**SCORE (TEAM\_ID, PID, RUNS\_MADE, OUT\_TYPE, B\_ID, BTEM\_ID)**

**BALLING (TEAM\_ID, B\_ID, OVERS, MAIDEN, RUNS\_GIVEN, WICKET\_TAKEN)**

In the above tables, B\_ID is the bowler id and refers to the P\_ID of PLAYER table. The SCORE table stores details about the scores of the batsmen and also stores details about how t hey got out. The BALLING table stores details about the bowlers.

1. Write DDLs to create the above table with necessary constraints.

CREATE TABLE TEAM\_MASTER(

TEAM\_ID INT PRIMARY KEY,

TEAM\_NAME VARCHAR2(10)

);

CREATE TABLE PLAYER(

TEAM\_ID INT,

PID INT,

PNAME VARCHAR2(15),

PBIRTHDATE DATE,

PRIMARY KEY(TEAM\_ID,PID),

FOREIGN KEY(TEAM\_ID) REFERENCES TEAM\_MASTER(TEAM\_ID)

);

CREATE TABLE SCORE(

BTEAM\_ID INT,

TEAM\_ID INT,

PID INT,

RUNS\_MADE INT,

OUT\_TYPE VARCHAR2(10),

B\_ID INT,

PRIMARY KEY(BTEAM\_ID),

FOREIGN KEY(TEAM\_ID, PID) REFERENCES PLAYER(TEAM\_ID, PID),

FOREIGN KEY(B\_ID) REFERENCES BALLING(B\_ID)

);

CREATE TABLE BALLLING(

B\_ID INT,

TEAM\_ID INT,

OVERS INT,

MAIDEN INT,

RUNS\_GIVEN INT,

WICKETS\_TAKEN INT,

PRIMARY KEY (B\_ID),

FOREIGN KEY(B\_ID, TEAM\_ID) REFERENCES PLAYER (PID,TEAM\_ID)

);

1. Display details about the bowler who took the maximum wickets.

SELECT \* FROM BALLING WHERE WICKET\_TAKEN=(SELECT MAX(WICKET\_TAKEN) FROM BALLING);

1. Which players are born in the month of October and have scored more than 25 runs?

SELECT \* FROM PLAYER WHERE EXTRACT(MONTH FROM PBIRTHDATE)=10 AND PID IN(SELECT PID FROM SCORE WHERE RUNS\_MADE>25);

1. Display details about players who bowled at the most three maiden overs.

SELECT \* FROM PLAYER WHERE PID IN (SELECT B\_ID FROM WHERE MAIDEN <4);

or

SELECT \* FROM PLAYER P JOIN BALLING B ON P.PID=B.B\_ID WHERE B.MAIDEN<4;

1. Display the total runs given by each bowler.

SELECT B.B\_ID, B.RUNS\_GIVEN FROM BALLING B JOIN PLAYER P ON B.B\_ID=P.PID;

1. Display team wise how many players got out with lbw, stumping, catch etc.

SELECT TEAM\_ID, OUT\_TYPE, COUNT(\*) TOTAL FROM SCORE GROUP BY (TEAM\_ID, OUT\_TYPE) ORDER BY TEAM\_ID;

1. Display details about players who are more than 25 years old.

SELECT \* FROM PLAYER

WHERE TRUNC(MONTHS\_BETWEEN(SYSDATE, PBIRTHDATE) / 12) > 25;

1. Display the bowlers who are also batsmen.

SELECT \* FROM PLAYER WHERE PID IN (SELECT PID FROM SCORE S JOIN BALLING B ON S.PID = B.B\_ID);

1. Create a read only view, which contains details about the team name and all its players.

CREATE VIEW TEAM\_DET AS SELECT \* FROM TEAM\_MASTER T JOIN PLAYER P ON T.TEAM\_ID=P.TEAM\_ID;

**CANDIDATE(CID,CNAME,CADDRESS, CBIRTH\_DT)**

**TEST(TID,TNAME,TOT\_MARKS,PASS\_MKS)**

**TEST\_CENTRE(TCID,LOCATION,MGR,CAPACITY)**

**TEST\_TAKEN(CID,TID,TCID,TEST\_DT,SCORE)**

1. Design the above tables with appropriate constraints.

CREATE TABLE CANDIDATE(

CID INT PRIMARY KEY,

CNAME VARCHAR2(20),

CADDRESS VARCHAR2(35),

CBIRTH\_DT DATE

)

CREATE TABLE TEST(

TID INT PRIMARY KEY,

TNAME VARCHAR2(15),

TOT\_MARKS INT,

PASS\_MKS INT

)

CREATE TABLE TEST\_CENTRE(

TCID INT PRIMARY KEY,

LOCATION VARCHAR2(25),

MGR VARCHAR,

CAPACITY INT

)

CREATE TABLE TEST\_TAKEN(

CID INT,

TID INT,

TCID INT,

TEST\_DT DATE,

SCORE FLOAT,

PRIMARY KEY(CID,TID,TCID),

FOREIGN KEY(TCID) REFERENCES TEST\_CENTRE(TCID),

FOREIGN KEY(TID) REFERENCES TEST(TID),

FOREIGN KEY(CID) REFERENCES CANDIDATE(CID)

)

1. Display the number of candidates who have appeared for each test at each test centre.

SELECT TID, TCID, COUNT(\*) FROM TEST\_TAKEN GROUP BY (TCID,TID) ORDER BY TCID;

1. Display candidate details for those candidates who have scored as many marks as Peter for tests conducted at Bangalore.

SELECT \* FROM CANDIDATE WHERE

CID IN (SELECT CID FROM TEST\_TAKEN

WHERE SCORE=(SELECT SCORE FROM TEST\_TAKEN

WHERE CID=(SELECT CID FROM CANDIDATE

WHERE CNAME='PETER')) AND

TCID=(SELECT TCID FROM TEST\_CENTRE WHERE LOCATION='BANGALORE'))

1. Display details about test centers where no tests have been conducted.

SELECT \* FROM TEST\_CENTRE TC WHERE NOT EXISTS(SELECT 1 FROM TEST\_TAKEN WHERE TC.TCID=TEST\_TAKEN.TCID);

1. Display details of candidates who have failed.

SELECT \* FROM CANDIDATE C JOIN TEST\_TAKEN TT ON C.CID=TT.CID JOIN TEST T ON TT.TID =T.TID WHERE SCORE<PASS\_MKS;

1. For tests, which have been conducted between 2-3-04 and 23-4-04, show details of the tests as well as the test centers.

SELECT T.TID, T.TNAME, T.TOT\_MARKS, T.PASS\_MKS, TC.TCID, TC.LOCATION, TC.MGR, TC.CAPACITY , TC.TEST\_DT FROM TEST T JOIN TEST\_TAKEN TT ON T.TID=TT.TID JOIN TEST\_CENTRE TC ON TT.TCID = TC.TCID WHERE TT.TEST\_DT BETWEEN '02-MAR-04' AND '23-APR-04'

1. How many candidates appeared in the ‘ORACLE FUNDAMENTALS’ test at Surat in the month of February?

SELECT COUNT(\*) FROM TEST\_TAKEN WHERE TID IN(SELECT TID FROM TEST WHERE TNAME='ORACLE') AND TCID IN (SELECT TCID FROM TEST\_CENTRE WHERE LOCATION='SURAT') AND EXTRACT(MONTH FROM TEST\_DT)=2

1. Display details about candidates who appeared for tests in the same month as the month in which they were born.

SELECT \* FROM CANDIDATE C JOIN TEST\_TAKEN T ON C.CID=T.CID

WHERE EXTRACT(MONTH FROM C.CBIRTH\_DT)= EXTRACT( MONTH FROM T.TEST\_DT);

1. Display the details about candidates who have scored the highest in each test, test centre wise.

SELECT TID, TCID, MAX(SCORE ) FROM TEST\_TAKEN GROUP BY TCID,TID ORDER BY TCID;

OR

SELECT C.CID, C.CNAME, T.TID, T.TNAME, TC.TCID, TC.LOCATION FROM CANDIDATE C JOIN TEST\_TAKEN TT

ON C.CID=TT.CID JOIN TEST T

ON TT.TID=T.TID JOIN TEST\_CENTRE TC

ON TT.TCID=TC.TCID

WHERE (TT.TID,TT.TCID,TT.SCORE) IN

(SELECT TID,TCID,MAX(SCORE) FROM TEST\_TAKEN GROUP BY TCID,TID)

ORDER BY TT.TCID,TID ;

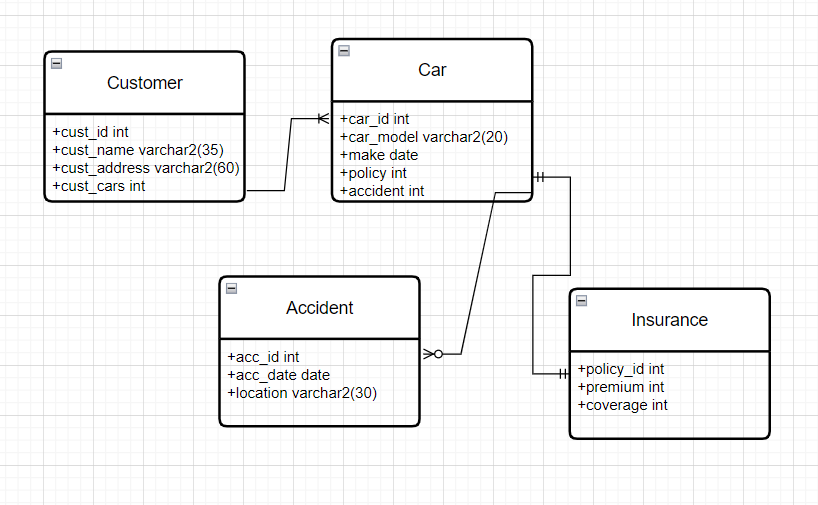
1. Design a read only view that has details about candidates and the tests that he has appeared for.

CREATE VIEW CANDIDATE\_TEST AS SELECT \* FROM CANDIDATE C JOIN TEST\_TAKEN TT ON C.CID = TT.CID;

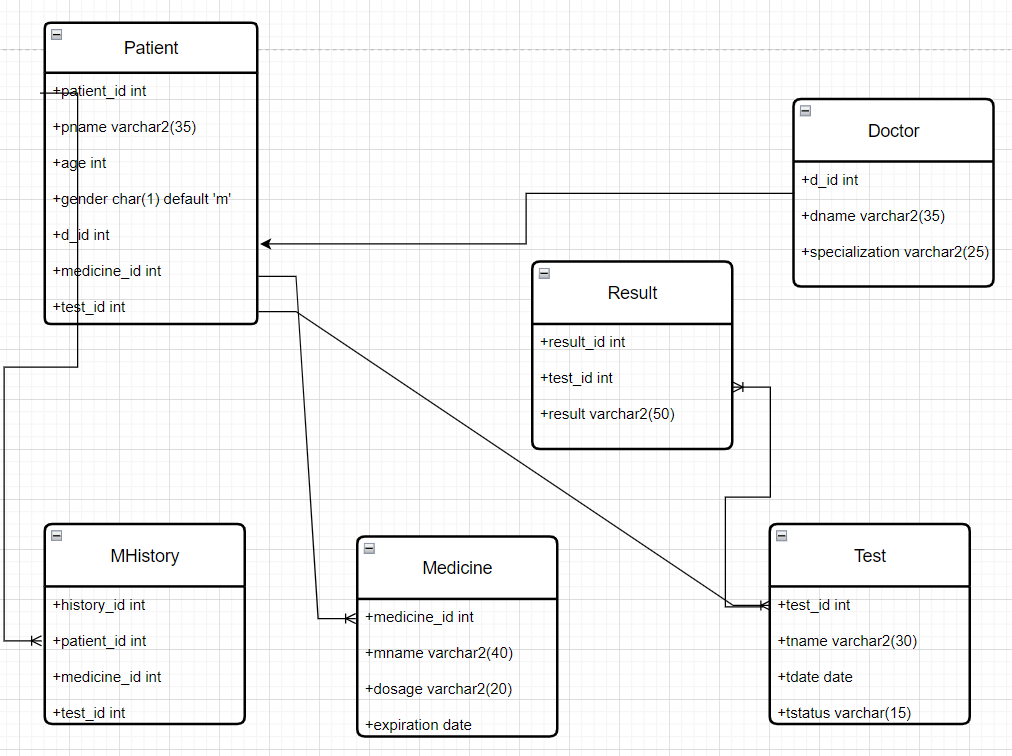
# Assignment 1c

# E-R Diagram

1. Construct an E-R Diagram for a car insurance company with a set of customers, each of whom owns a number of cars. Each car has a number of recorded accidents associated with it. The details about the insurance policies of the cars as well as the details of the accidents are also stored.



1. Construct an E-R Diagram for a hospital management system. The details of the patients and the doctors will be required as well as the details of the tests and its results conducted on a patient. The patient history is also stored for future reference. Everything related to the medicines prescribed is also required.



1. Draw an E-R diagram to depict the different activities that go on in a shopping mall. Take Vadodara Central as a case study and try to design your tables if you were to computerize its activities. Mention your scope very clearly.

