9.4.1 page (440)

PSM - persistent stored modules, each dbms has an extension

a) Given the name of a movie studio produce the net worth of it's president

CREATE FUNCTION presidentNetWorth(IN studioName varchar(20)) RETURNS INT

DECLARE PresNetWorth INT;

SELECT netWorth into PresNetWorth

FROM studio, movieExec

where studio.name = studioName AND presC# = cert#;

return PresNetWorth;

b) given name and address. return: 1-movie star not executive, 2-execuive not a star, 3-both, 4-neither

CREATE FUNCTION personStatus(IN name1 varchar(20), IN address1 varchar(255)) RETURNS INT

DECLARE starStatus BOOL;

DECLARE execStatus BOOL;

DECLARE status INT;

SET starStatus = FALSE;

SET execStatus = FALSE;

SELECT TRUE INTO starStatus

IF EXISTS (SELECT \* FROM MovieStar

WHERE name=name1 AND address=address1);

END IF;

SELECT TRUE INTO execStatus

IF EXISTS (SELECT \* FROM MovieExec

WHERE name=name1 AND address=address1);

END IF;

IF starStatus = TRUE and execStatus = FALSE SELECT 1 INTO status;

ELSE IF execStatus = TRUE and starStatus = FALSE SELECT 2 INTO status;

ELSE IF starStatus = TRUE and execStatus = TRUE SELECT 3 INTO status;

ELSE starStatus = FALSE and execStatus = FALSE SELECT 4 INTO status;

END IF;

RETURN status;

e)(extra) Given an address, find the name of the unique star with that address if there is exactly one, and return null if there's more or none.

CREATE FUNCTION starFromAddress(IN starAddress varchar(255)) RETURNS varchar(255)

DECLARE starsName varchar(255);

IF 1 = (SELECT COUNT(name) FROM MovieStar where address = starAddress)

THEN RETURN (SELECT name FROM MovieStar where address = starAddress)

ELSE RETURN NULL;

END IF;

f) (extra) given name of star, delete them from MovieStar and delete all their movies from StarsIn and Movies

CREATE PROCEDURE deleteAllAboutStar(IN starName1 varchar(40))

DELETE FROM Movies WHERE Movies.title=StarsIn.MovieTitle AND starName=starName1;

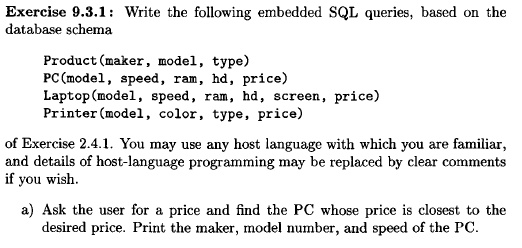
DELETE FROM StarsIn WHERE starName=starName1;

DELETE FROM MovieStar WHERE name=starName1

Page 449(pdf) 412(book)



Page 427(pdf) 390(book)



#inlcude sqlcli.h

#include <stdio.h>

void main () {

/\*declaring needed variables\*/

int i;

SQLHENV myEnv;

SQLHDBC myCon;

SQLHSTMT execStat;

SQLINTEGER queryCol1, queryCol2, queryCol3 queryResultInfo;

SQLRETURN errorCode1, errorCode2, errorCode3;

/\*setting up database\*/

errorCode1 = SQLAllocHandle(SQL\_HANDLE\_ENV, SQL\_NULL\_HANDLE, ftmyEnv);

if( !errorCode1){

errorCode2 = SQLAllocHandle(SQL\_HANDLE\_DBC, myEnv, fanyCon);

}

if( ! errorCode2){

errorCode3 = SQLAllocHandle(SQL\_HANDLE\_STMT, myCon, ftex e cS tat);

}

/\*prompting user, then scanning input\*/

printf(“Enter a price for a PC:”);

scanf(“%d”, i);

/\*preparing, and then executing statement saved in str

This is a big sql query but basically it does everything we

need for this exercise, if there is a result it will be the

model number, maker and speed of the pc with the price closest

to whatever is specified by the user.\*/

SQLPrepare(execStat, "SELECT A.model, Product.maker, A.speed

FROM (SELECT model

FROM (SELECT model, speed, ABS(price - ?) AS diff

FROM PC) AS A

WHERE diff <= (SELECT MIN(ABS(price - ?)) AS diff

FROM PC)) AS A, Product

WHERE A.model = Product.model", SQL\_INTS);

/\*binding parameters to question marks in the statement,

for some reason the entirety of SQLBindParameter is

never explained, so I only inputted the essential stuff?!?\*/

SQLBindParameter(execStat, 1, ..., i, ...);

SQLBindParameter(execStat, 2, ..., i, ...);

SQLExecute(execStat);

/\*putting results into variables\*/

SQLBinCol(execStat, 1, SQL\_INTEGER, &queryCol1, sizeof(queryCol1), &queryResultInfo);

SQLBinCol(execStat, 2, SQL\_INTEGER, &queryCol2, sizeof(queryCol2), &queryResultInfo);

SQLBinCol(execStat, 3, SQL\_INTEGER, &queryCol3, sizeof(queryCol3), &queryResultInfo);

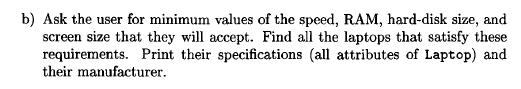
/\*fecthing result and printing result\*/

while(SQLFetch(execStat) != SQL\_NO\_DATA){

printf("Model: %d, Maker: %d, Speed: %d\n", queryCol1, queryCol2, queryCol3);

}

}



#inlcude sqlcli.h

#include <stdio.h>

void main () {

/\*declaring needed variables\*/

int userSpeed, userRam, userHd, userScreen;

SQLHENV myEnv;

SQLHDBC myCon;

SQLHSTMT execStat;

SQLINTEGER queryCol1, queryCol2, queryCol3, queryCol4, queryCol5, queryCol6 queryResultInfo;

SQLRETURN errorCode1, errorCode2, errorCode3;

/\*setting up database\*/

errorCode1 = SQLAllocHandle(SQL\_HANDLE\_ENV, SQL\_NULL\_HANDLE, ftmyEnv);

if( !errorCode1){

errorCode2 = SQLAllocHandle(SQL\_HANDLE\_DBC, myEnv, fanyCon);

}

if( ! errorCode2){

errorCode3 = SQLAllocHandle(SQL\_HANDLE\_STMT, myCon, ftex e cS tat);

}

/\*prompting user, then scanning input\*/

printf(“Enter a price for a PC:”);

scanf(“%d”, i);

/\*preparing, and then executing statement.

This is a big sql query but basically it does everything we

need for this exercise, if there is a result it will be the

model number, maker and speed of the pc with the price closest

to whatever is specified by the user.\*/

SQLPrepare(execStat, "SELECT \*

FROM Laptop

WHERE speed >= ? AND

ram >= ? AND

hd >= ? AND

screen >= ?", SQL\_INTS);

/\*binding parameters to question marks in the statement,

for some reason the entirety of SQLBindParameter is

never explained, so I only inputted the essential stuff?!?\*/

SQLBindParameter(execStat, 1, ..., userSpeed, ...);

SQLBindParameter(execStat, 2, ..., userRam, ...);

SQLBindParameter(execStat, 3, ..., userHd, ...);

SQLBindParameter(execStat, 4, ..., userScreen, ...);

SQLExecute(execStat);

/\*putting results into variables\*/

SQLBinCol(execStat, 1, SQL\_INTEGER, &queryCol1, sizeof(queryCol1), &queryResultInfo);

SQLBinCol(execStat, 2, SQL\_INTEGER, &queryCol2, sizeof(queryCol2), &queryResultInfo);

SQLBinCol(execStat, 3, SQL\_INTEGER, &queryCol3, sizeof(queryCol3), &queryResultInfo);

SQLBinCol(execStat, 4, SQL\_INTEGER, &queryCol4, sizeof(queryCol4), &queryResultInfo);

SQLBinCol(execStat, 5, SQL\_INTEGER, &queryCol5, sizeof(queryCol5), &queryResultInfo);

SQLBinCol(execStat, 6, SQL\_INTEGER, &queryCol6, sizeof(queryCol6), &queryResultInfo);

/\*fecthing result and printing result\*/

while(SQLFetch(execStat) != SQL\_NO\_DATA){

printf("Model: %d, Speed: %d, Ram: %d, Hd: %d, Screen: %d, Price: %d\n", queryCol1, queryCol2, queryCol3, queryCol4, queryCol5, queryCol6);

}

}

Exercise 9.6.1 do 9.3.1 in JDBC look at jdbc\_exercise\_examples.java

