Database Practical 5 1999-2000

Objectives To set up relationships between tables, and exploit them in queries and forms. To understand referential integrity.

(a) Copy the database **e:\examples\jgb\Oscars.mdb** to the **d:\docs** directory. **Open** the database **Oscars** in **Access.** Read through *Creating, and Designing Tables* => *Defining relationships and Setting Referential Integrity Options* => ***About relationships in an Access database.***

(b) Have a look at (the design and datasheet views of) the **Films and Directors** and **Nominations for Acting** tables, and the **Films with Cast** query of the **Oscars** database. In particular, you should inspect the design view of the **Films with Cast** query, which shows the join produced by the relationship between the two tables it uses.

(c) Your objective now is to produce a table to record the venues at which some of the films are currently being shown. Before you start, you should sketch the structure described in **(d), (e)** and **(f)** as an Entity-Relationship diagram. Note particularly the assumption in **(f).** If you are unsure about it, discuss your Entity-Relationship diagram with a demonstrator.

(d) Create a new table with the fields **Venue Name, City or Town** and **Film ID** (and a few more if you wish). The **Film ID** field should be of type **number** and Field Size **long integer.** (It must match the **Film ID** field in the **Films and Directors** table; matching fields must be of the same data type, except that a long integer number is allowed as a match for an Autonumber.) When you have designed your table, save the design.

(e) Read through *Creating, and Designing Tables* => *Defining relationships and Setting Referential Integrity Options* => ***Define relationships between tables.*** Now open the **Relationships Window.** You should adjust the position of the tables so that you can see all the relationships clearly. Add your new table to the window. Set up an appropriate relationship between your new table and the **Films and Directors** table. Read through *Creating, and Designing Tables* => *Defining relationships and Setting Referential Integrity Options* => ***What is referential integrity.*** You should **Enforce Referential Integrity** *now*, before you enter any data. ***Don't*** set the **Cascade Update …** and **Cascade Delete…** checkboxes.

(f) Add a few records to your new table. Use your imagination for venues if you wish, but the **Film ID** should come from those in the **Films and Directors** table. Assume that each venue is currently showing only one film, but a film may be showing at several venues; your records should exemplify this.

(g) Create a query to show, in alphabetical order of venue, the venue, film title, director's name and year of nomination for the films in your table listing current screenings. (You will need to add both the related tables from **(e)** to the query; the relationship you set up in **(e)** will automatically be used to relate them.)

(h) Read through *Working with Forms* => *Creating Multiple-Table or Linked Forms (Subforms)* => ***Subforms: What they are and how they work.*** Use your query to set up a form showing a film (title) and its director, together with a list of the venues at which it is currently showing. (You may use a **Wizard.**)

(i) Did you remember to print sample copies of what you produced for your practical file?

(j) The following experiment is to illustrate how referential integrity works.

- Start a new blank database. Create two tables, **Employees** and (company) **Cars** with the fields shown below. ***Create the tables with no primary keys and no indexes.*** **Employees** holds information about people. **Cars** holds information about company vehicles, including the person to whom the car is assigned (recorded by quoting their surname).

Employees
Surname
Post

Cars
Registration
Person

- Try to set up a relationship between **Surname** and **Person,** enforcing referential integrity. What happens? Why?
- Set **Surname** as a primary key on **Employees**. Now set up a relationship between **Surname** and **Person,** enforcing referential integrity.
- Open both tables. Add records (Surname, Post) of five employees. Now try to allocate a car to a non-existent employee. What happens, and why? Add four valid records to the **Cars** table.
- Delete a car – see that you can. Is this what you expect? Why?
- Try to delete an employee with a car – see that you can't. Is this what you expect? Why?
- Delete an employee without a car – see that you can. Is this what you expect? Why?

(k) ***Before signing off your practical,*** the demonstrator will ask to see your sketch of the Entity-Relationship diagram from **(c),** your form with subform (and perhaps the query on which it is based), and may ask you questions to test your understanding of the referential integrity experiment in **(j).**

(l) If you want to try out your queries on the database of West End productions discussed in class, you will find a copy of the "this is one I made earlier" database I showed in the lecture at **e:\examples\jgb\West_End.mdb.**

(m) If you have time to spare, explore the **Oscars** database.