N5 Computing Science Revision Notes

# Database Design & Development

|  |
| --- |
| **Analysis** |
| ***End-user requirements***  The end users are the people who are going to be using the database. Their requirements are the tasks they expect to be able to do using the database.  ***Functional requirements***  Processes and activities that the system has to perform. Information that the system has to contain to be able to carry out its functions.  **Example**  A used-car dealership has six showrooms in different locations across Scotland. It wants to create a relational database to store details of cars owned by the company and details of each of their showrooms. The database will allow sales staff to view details of specific cars and to see which showroom the cars are located in.  The developers have asked some of the sales staff what they would like to see in the database. Here are a few of the comments they made.    *End-user requirements*  Sales staff should be able to display details of cars, by performing a number of different searches using:   * car make and model * car colour * range of car prices * type of fuel used   Search results should display:   * car make * car model * car price * car fuel * car mileage * branch address * branch telephone number   Users should be able to sort the search results in ascending order of mileage and ascending order of price.  *Functional requirements*  The relational database will have two tables; one for car details and one for branch details:   * Each table will require a suitable primary key field. * A foreign key will be used to link the two tables.   Additional fields will be needed for:   * car make * car model * car colour * car fuel — diesel, petrol or electric * car price * car mileage * branch address * branch telephone number   Simple and complex queries will be used to search the database.  A complex sort will be used to order the query results. |

|  |
| --- |
| **Design** |
| ***Data Protection Act 1998***  The Data Protection Act aims to protect anyone who has personal information stored on a computer system. It outlines a set of responsibilities for the person who stores the data.  ***Entity Relationship Diagrams***  ***Data Dictionary***  Used to describe the structure of a database and to indicate the properties of each attribute (field) needed to define the entities (tables).  ***Query***  ***Example***  The SportsStats website uses a relational database to store details of Scottish sports teams and players in two separate tables called Player and Team.   * All players have a unique Player ID that has 4 characters * Players are given a star rating between 1 and 5 * The website features a limited number of sports: basketball, handball, hockey and netball   Sample data stored in each table is shown below.  Player table   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Player ID** | **Team ID** | **Star Rating** | **First Name** | **Last Name** | **Date of Birth** | | L18C | 111 | 4 | Lindy | Osborne | 14/01/1990 | | F19F | 113 | 3 | Fred | Freddricks | 30/07/1987 | | Y01D | 131 | 5 | Yasmine | Davies | 22/11/1992 |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Team table | | | | | | | **Team ID** | **Team Name** | **Sport** | **Manager** | **Coach** | **Home Town** | | 111 | West Stars | Hockey | Chris Davidson |  | Paisley | | 112 | Killie Shooters | Basketball |  | Liz Smillie | Ardrossan | | 113 | Jumpin Jacks | Basketball | Dave Ford | Ali Mustapha | Dunbar |   A data dictionary for the SportsStats database is shown below.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Player table** | | | | | | | **Field** | **Key** | **Type** | **Field Length** | **Reqd** | **Validation** | | Player ID | PK | Text | 4 | Y | Presence Check and Length Check | | Team ID | FK | Number | 3 | Y | Lookup from Team table | | Star Rating |  | Number | 1 | Y | Range Check | | First Name |  | Text | 20 | Y |  | | Last Name |  | Text | 20 | Y |  | | Date of Birth |  | Date | 10 | Y |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Team table** | | | | | | | **Field** | **Key** | **Type** | **Field Length** | **Reqd** | **Validation** | | Team ID | PK | Number | 3 | Y | Presence Check | | Team Name |  | Text | 20 | Y |  | | Sport |  | Text | 20 | Y | Restricted Choice | | Manager |  | Text | 40 | Y |  | | Coach |  | Text | 40 | Y |  | | Home Town |  | Text | 20 | Y |  | |

|  |
| --- |
| **Implementation** |
| ***Relational Databases***  ***SQL Operations***  SQL stands for ‘Structured Query Language’. SQL is a special purpose programming language for storing, manipulating and retrieving data in relational databases.  Example  The Scottish Handball League use a relational database to store details of teams and players in two separate tables called Team and Player. The structure of these tables is shown below:    A sample record stored in each table is shown below:    **Searching**  The **SELECT** statement is used to decide which fields should be displayed. The statement is followed by the fields, separated by commas.  The **FROM** clause states the names of the database table(s) that are needed in the query.  The **WHERE** clause states the criteria that must be met. This clause is followed by the field name, an operator (<, >, =) and the information inside inverted commas if it is text.  SELECT fieldName1, fieldName2, fieldName3, etc  FROM tableName  WHERE fieldName = data;  ***Example 1*** To search the database to display the town, contact person, e-mail address and district for all teams in the west district, you would write the following SQL.    **Logical operators** (AND, OR) can be used to create complex criteria.  ***Example 2*** To search the database to display the surname, position, and division of all players who either play in the right wing or centre position, you would write the following SQL.    **EQUI-JOIN** between tables  If a search involves displaying data found in two linked tables the SQL clause must state the link. An **EQUI-JOIN** is added to the **WHERE** statement stating that the primary and foreign key values in both tables must match.  SELECT fieldName1, fieldName2, fieldName3  FROM tableName1, tableName2  WHERE tableName1.fieldNamePK = tableName2.fieldNameFK  AND fieldName = data;  ***Example*** 3 To search the database to display the full name and town of any players who play for a handball team based in Paisley, you would write the following SQL. |

|  |
| --- |
| **Testing** |
| ***SQL Operations*** |

|  |
| --- |
| **Evaluation** |
| ***Fitness for Purpose***  ***Accuracy of Output*** |