|  |  |
| --- | --- |
| **Databases**  Diploma in IT / DS / CSF  Year 1 (2023/24) Semester 2 | Week 7 |
| **2** hours |
| **TUTORIAL 6**  **Relational Theory / Relational Model / Data Dictionary** | |

**Learning Objectives**

1. Learn the basic theories of Relational Model
2. Learn the basics of Data Dictionary
3. Learn how to construct data dictionary information for database tables

**Task 1 – Apply Basic Theories of Relational Model**

In DB Design process, a designer first creates the ER model by studying the business, processes and information requirements. The ER model is then mapped into relations via Logical DB design and these relations are finally created into DB tables in implementation.

We will examine the Relational Model this week and learn to map ER model into relations in activities next week.

* 1. **Basic Terms in Relational Model** *(refer to pp 1-7 to 1-12)*

The key terms used in Relational Model include: relation, attribute, degree, cardinality, tuple and domain.

**Relation** – a named table that consists of fixed number of columns and any number of rows. Each relation’s name must be unique within the database.

**Attribute** – a named column in a relation. Name of each attribute must be unique within a relation.

**Degree** – the number of attributes in a relation.

**Cardinality** – number of tuples (or rows) in a relation.

**Tuple** – a row in a relation (just like an instance in an entity)

**Domain** – set of allowable values for an attribute (e.g. gender has value ‘M’ or ‘F’)

**1.2 Apply Relational Model Terms**

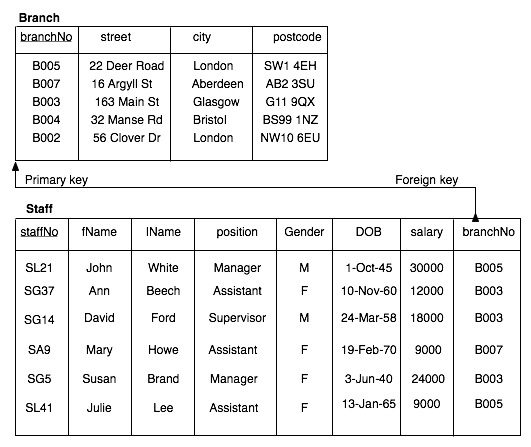
Use Relational Model terms on the following relations:

Figure 1 – Sample Relations

*Source: Database Systems, A Practical Approach to Design, Implementation and Management, by Thomas Connolly, Carolyn Begg*

1. Name the relations: Branch and Staff
2. Identify all of the attributes for Branch relation: branchNo, street, city and postcode
3. What is the degree for Branch relation? 4
4. What is the cardinality for Branch relation? 5
5. What is the domain for City attribute in Branch relation? Name of cities
6. Identify anyone tuple in Branch relation: “B005, 22 Deer Road, London, SW1 4EH”

**1.3 Properties of Relations** *(refer to pp 1-13)*

In Relational Model, the relations have to observe these properties:

1. A relation has a name that is different from other relations in the same database
2. Each attribute name must be distinct (or unique) within a relation
3. Each attribute is allowed to store one value (or atomic value)
4. The values of the attribute must be from the same domain
5. Each tuple in a relation is distinct (in other words, tuples cannot be duplicated)
6. The order of attributes in a relation is not important
7. The order of tuples in a relation is not important

**1.3.1** Answer the following questions by referring to relations in Fig 1. For each question, you should answer whether the action is allowed or not allowed. If it is not allowed, explain which property in a relation that may be violated (in other words, which relation property is not followed).

1. Add a new attribute “Manager Name” to Branch relation

Allowed. It is a new attribute and the name is not duplicate.

1. Add the following tuple to Branch relation: “889Z, 1 Orchard Road, Liverpool, AB3 4TU.”

Not allowed, wrong domain is used for branchNo. Domain value should be in the format of B999.

1. Add a new attribute “Nationality” to Staff relation

Allowed. A new attribute and the name is not a duplicate

1. John White is given another appointment, “Acting Senior Mgr”. Add this appointment to his position.

Not allowed. It violates the property that an attribute must have a single value or an atomic value.

1. Management has decided to open branches overseas in Europe. A new relation named “Branch” is created to hold information on these overseas branches.

Not allowed. It violates the property that a relation name must be unique in an entity.

**1.4 Integrity Rules in Relational Model** *(refer to pp 1-20 to 1-24)*

There are 2 integrity rules in Relational Model: **Entity Integrity and Referential Integrity**.

Integrity rules are important in database as this ensures that your data is valid. For example, integrity rule will ensure that a customer order is placed by an existing or valid customer; otherwise, you will not be able to trace who actually places this order.

1. What is Entity Integrity? The primary key in a relation must be unique and cannot be NULL.
2. What is Referential Integrity? The FK value in relation must matched the corresponding PK value in another relation (or same relation) or it carries a NULL value.
3. Are the following actions applied to relations in Fig 1 allowed?
   1. Add a new tuple to Staff relation with these values:

“SL33, Margaret, Day, Supervisor, F, 1-Jun-60, 1500, B987”

Not allowed. There is FK B987 in the tuple but there is no corresponding branchNo in the Branch. It violates Referential Integrity.

* 1. Close the branch “B005” in London as it is not doing well. Remove this tuple from Branch relation.

Not allowed. It violates the Integrity as there are 2 existing staff tuples that have FK of B005. To resolve this, we need to either reassign the branchNo of the affected staff to another branch or set their branchNo value to NULL before proceeding to remove the given branch tuple.

* 1. Open a new branch at Manchester. Management has not decided on the branch code and left the branch code as “NULL”.

Not allowed. It violates Entity Integrity rule. The PK cannot be NULL value.

**Task 2 – Understand the Basics of Data Dictionary (OPTIONAL)**

In designing a database, the designer has to grapple with a host of information: database tables, their attributes or columns, the data type for attributes and constraints. What tool can a database designer use to help him organise and retrieve the relevant information in a database? The tool is the **Data Dictionary**.

Just like in everyday language usage, you refer to a dictionary when you need to check the word spelling, meaning or origin. A good DBMS (database management software) has an integrated data dictionary component to provide information and answers to the database designer or application programmer on the database details.

**2.1 What are the terms used to refer to Data Dictionary?**

“DBMS maintains a **system catalog**, a **repository** of information containing data that describes the data in a database” Other terms used to refer to system catalog are Data Dictionary or Data Directory.

**2.2 Why use a Data Dictionary?**

As mentioned in the preamble, a database designer grapples with thousands of pieces of information describing a database – database tables, their columns, the data types, the constraints and other relevant information. It is important that this information on database (or commonly referred to as *metadata*) is integrated into a DBMS software for easy reference.

**2.3 What are the key information items in a Data Dictionary?**

Let’s look at the basic information in a data dictionary. Below is an extract of Book Table from NP40Book:

**Book**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attribute** | **Description** | **Data Type** | **Constraint** | **Null Value** |
| ISBN | Unique identifier of each book | char(10) | Primary key | No |
| Title | Title of Book | Varchar(200) |  | No |

**Attribute** – name assigned for each attribute in the table (this comes from the attribute attached to the entity when you are modelling the database). Attribute name must be unique in a table.

**Description** – a brief explanation or description of this attribute

**Data Type** – the data type assigned for this attribute

**Constraint** – defines whether this attribute contains a constraint like Primary Key, Foreign Key or contains certain permissible values (such as gender must be ‘M’ or ‘F’)

**Null Value** – is it permitted for the attribute to hold a Null value at the point of creating the database record? Values must be “Yes” or “No”

**Task 3 - Exercise on Data Dictionary Terms**

(a) Why must the attribute name be unique in a database table?

Ans: To ensure that there is no ambiguity when referring to a particular column in a table (this also confirms the property of a relation, namely, that there cannot be duplicate column names)

(b) Is it permitted to have the same attribute name in another table?

Ans: Yes. It is okay to have the same attribute name in another table. In fact, database designers usually use the same attribute name to imply FK relationship.

(c) Besides char and varchar, what are the other commonly used data types?

Ans: int, float, smalldatetime, money, smallmoney

(d) What is the difference between char(20) and varchar(20)?

Ans: char(20) sets aside 20 characters of a storage space for attribute value regardless of whether all the 20 characters or space is used up or not.

varchar(20) allows users to enter up to 20 characters but will use up less storage space if the 20 characters is not completely utilised. As a general rule, use the varchar datatype if the attribute value length is likely to vary across different records (example, Name of person, address, product description, etc)

(e) When will you define an attribute as data type number (e.g. Int, decimal) instead of char?

Ans: Define a number or numeric data type if this attribute value is to be use for calculation. Example of this usage may be quantity, price, weight, etc. char datatype only stores the string data type and strictly should not be use for calculation purposes.

(f) Why is it important to define Constraints in a data dictionary?

Ans: Constraints help to ensure that the values stored in the database are correct and confirmed to certain rules. Example of constraints will be Entity Integrity rule for PK and Referential Integrity rule for FK. Other constraints ensure that other domain values are adhered to (gender may have ether ‘F’ or ‘M’, price range may be from $1 up to $50 if the item pricing will not exceed $50).

(g) When Null value is defined as “No”, what does it mean? State an example.

Ans: This means that at the point of record creation, a value must be entered for this attribute. The value entered must also confirm to the constraints specify for this attribute. For example, if the constraint is PK, then this value must be unique compared to existing attribute value for this table. Example of NULL value set to ‘No’ -a PK value. Another example, when entering customer details, we will expect to have a value for customer name and customer contact.

(h) When Null value is defined as “Yes”, what does it mean? State an example.

Ans: This means that at the point of the record creation, a value need not be entered for this attribute. When this happens, the record will be created with this record carrying the NULL value.

Examples of usage, the categorisation of the item may not be decided. For example, when we categorised a video, we have yet to decide which category the video should be categorised. Another example, when product items are brought in for retail, the manager may have yet to decide the selling price and this value will be set to NULL before updating it later.

**Task 4 – Define Items for Uniqlo’s receipt**

Recall the activity in week 2 when we examined the documents and identified the attributes in those documents.

Re-examine the Uniqlo’s receipt document and complete the table with information for these identified attributes listed below. You should carefully examine the attributes, their format and length before filling up the table with items.

****

**Uniqlo’s Receipt**

| **Attribute** | **Description** | **Data Type** | **Constraint** | **Null Value** |
| --- | --- | --- | --- | --- |
| Receipt\_No | Number assigned to receipt | Char(22)  E.g. 2013040617270003050197 | Primary Key | No |
| Receipt\_Date | Date that receipt is generated | DateTime  E.g. 06/04/2013 |  | No |
| Receipt\_Time | Time that receipt is generated | DateTime  E.g. 17:27 |  | No |
| Store\_No | Store number of store where purchase is made | Varchar(4)  E.g. “0003” |  | No |
| POS | Point of sale when the sale is made | Int  E.g. “5” |  | No |
| Employee\_No |  |  |  |  |
| Tran\_No |  |  |  |  |
| Employee\_Name |  |  |  |  |
| Store\_Name |  |  |  |  |
| Store\_Address |  |  |  |  |
| Store\_Phone\_No |  |  |  |  |
| Item\_Code |  |  |  |  |
| Item\_Name |  |  |  |  |
| Item\_Qty |  |  |  |  |
| Item\_Price |  |  |  |  |
| Total\_Items |  |  |  |  |
| Receipt\_Amt |  |  |  |  |
| Payment\_Mode |  |  |  |  |
| Paid\_Amt |  |  |  |  |
| Change\_Amt |  |  |  |  |
| GST\_Amt |  |  |  |  |

**Task 5 – ER Model Checkpoint Consultation (OPTIONAL – Do this in class if you have time)**

1. Finalized ER model. It should contain at least 6 entities identified from the core business processes and should support the additional non-trivial queries that you have identified.
2. List of assumptions made and/or constraints of your ER model.

**Submit the above documents to CheckPoint Submission in BrightSpace in week 7.**

