ANALYSIS PRESENTATION

PROGRAMMING FOR DATA ANALYSIS

Designing an Efficient Database Schema for Organizing and Managing Items with Categories and Descriptors

Presented By: Peter Ibeabuchi

Problem Statement

We have a collection of items with different categories and descriptors, Our goal is to design a database schema and tables that can efficiently organize and manage this information.

2 Tables vs 6 Tables

2 Tables Approach	6 Tables Approach
The data set is small, only 30 items to describe. So, it's easy to do CRUD operations. We just need to target one table i.e. "Items" table	Hard to do CRUD operation to get data from whole data sets we need to retrieve data from all six tables
Primary Key Foreign Key relationship	No Primary Key Foreign Key relationship. Means longer and repetitive queries
We have common descriptors for all items. So why repetition in data and tables creation. Need only four queries to create tables and insert data	Need 12 queries to create and insert data. Six for table creation and six for data insertion
Simple queries, less complex joins. Easy to maintain queries. Easy to understand and easy to tweak queries to get desired result	Longer queries and multiple queries to use to data common data from whole dataset

Category Table

Column	Data Type	Reason
category_id	Smallint	The category_id is a unique(number) identifier for each category. Smallint provides a more suitable data type for representing a moderate number of categories. Alternatively, the integer data type can be used.
category_name	varchar(30)	The name of each categories are typically characters and strings. The varchar() data type provides a suitable approach to store string variables in the database. We chose a text length of 30, to accommodate each category names.

Items Table

Column	Data Type	Reason
item_id	Integer	We only have 6 categories so Smallint is the best choice. That's also the Primary Key for the table with auto increment.
category_id	Smallint	Foreign Key Column so data type should match.
item_name	Varchar (20)	Used Varchar datatype because item_name should be a string with some special characters and length based on the input values.
color	Varchar (20)	Value for the color will be in strings so we have chosen varchar as datatype.
Size	Varchar (20)	The data is in different categories for the safer side we choose varchar as datatype.
is_edible	Char (1)	We chose a char datatype with length one because we are going to use Y for Yes and N for No.
is_wearable	Char (1)	As in the previous column we used a char datatype with length one because here we used Y for Yes and N for No.
type	Varchar (20)	Value for this column will be in strings with some special character so we used varchar datatype.

Tables Creation and Data Insertion

Using the “CREATE” query, we created both tables(Category and Item), with the assigned column names and the respective constraints(PK and FK)

Also, using the “INSERT” query, all items identified in the image are inserted into the item tables with their various descriptions as the are obsereved. In total, 30 records were entered into the items table.

```
-- Inserting Item data into table
INSERT INTO items
(category_id, item_name, color, size, is_edible, is_wearable, type)
VALUES
(3, 'Water Melon', 'Green', 'Big', 'Y', 'N', 'Food'),
(3, 'Honey Jar', 'Yellow', 'Medium', 'Y', 'N', 'Food'),
(1, 'Long Shoes','Brown','Long',    'N',    'Y', 'Shoes'),
(2, 'Hand Bag'  ,'Purple',  'Medium',   'N',    'Y', 'Bag'),
(2, 'Back Pack' ,'Pink',    'Medium',   'N',    'Y', 'Bag'),
(2, 'Umberalla' ,'Blue',    'Medium',   'N',    'N', 'Accessory'),
(3, 'Donut' ,'Brown',   'Medium',   'Y',    'N', 'Food'),
(6, 'Dress',    'Black',    'Small',    'N',    'Y', 'Cloth'),
(3, 'Mushroom', 'Yellow',   'Small', 'Y', 'N', 'Food'),
(3, 'Strawberry Juice', 'Red',  'Big', 'Y', 'N', 'Food'),
(3, 'Chocolate Cake',   'Brown',    'Big', 'Y', 'N', 'Food'),
(2, 'Shopping Bag', 'Pink', 'Big'   , 'N', 'N', 'Bag'),
(6, 'Gloves',   'Red',  'Big', 'N', 'Y', 'Cloth'),
(3, 'Redish',   'Red',  'Small','Y', 'N', 'Food'),
(6, 'Socks',    'Blue', 'Small' , 'N', 'Y', 'Cloth'),
(3, 'Chocolate Caks','Brown',   'Big', 'Y', 'N', 'Food'),
(2, 'Shopping Bag', 'Pink'  ,'Big'  , 'N', 'N', 'Bag'),
```

```
CREATE table category (
  category_id smallint PRIMARY KEY IDENTITY(1,1),
  category_name varchar(30));

CREATE TABLE items (
  item_id integer PRIMARY KEY IDENTITY(1,1),
  category_id smallint,
  item_name varchar(20),
  color varchar(20),
  size varchar(20),
  is_edible Char(1),
  is_wearable Char(1),
  type varchar(20)
);
-- Foreign Key
ALTER TABLE items  WITH CHECK ADD  CONSTRAINT FK_Item_Category_ID FOREIGN KEY(category_id)
REFERENCES category (category_id)
GO
```

ITEM_ID	CATEGORY_ID	ITEM_NAME	COLOR	SIZE	IS_EDIBLE	IS_WEARABLE	TYPE
3	1	Long Shoes	Brown	Long	N	Y	Shoes
4	2	Hand Bag	Purple	Medium	N	Y	Bag
5	2	Back Pack	Pink	Medium	N	Y	Bag
8	6	Dress	Black	Small	N	Y	Cloth
15	6	Socks	Blue	Small	N	Y	Cloth
18	6	Muffler	Green	Medium	N	Y	Cloth
21	1	Casual Shoes	Yellow	Meduim	N	Y	Shoes
22	1	Ice Shoes	Green	Medium	N	Y	Shoes
26	2	Back Pack	Brown	Big	N	Y	Bag
34	1	Long Shoes	Brown	Long	N	Y	Shoes
35	2	Hand Bag	Purple	Medium	N	Y	Bag
36	2	Back Pack	Pink	Medium	N	Y	Bag
39	6	Dress	Black	Small	N	Y	Cloth
46	6	Socks	Blue	Small	N	Y	Cloth
49	6	Muffler	Green	Medium	N	Y	Cloth
52	1	Casual Shoes	Yellow	Meduim	N	Y	Shoes
53	1	Ice Shoes	Green	Medium	N	Y	Shoes
57	2	Back Pack	Brown	Big	N	Y	Bag

Q3: Filter for all wearable items that are not red.

SELECT * FROM items WHERE is_wearable = 'Y' AND Color != 'Red';

The query retrieves all the wearable items from the items table where item colors other than Red.

Methods Used:

SELECT – To retrieve the data from the Database.

FROM – Identifies the tables from which the data should be retrieved.

WHERE – Used to filter out the data based on the given condition.

AND – Used to filter records based on more than one condition (both conditions should be satisfied)

Q4: Filter for all edible items that are yellow.

item_id	category_id	item_name	color	size	is_edible	is_wearable	type
2	3	Honey Jar	Yellow	Medium	Y	N	Food
9	3	Mushroom	Yellow	Small	Y	N	Food
25	3	Pumpkin	Yellow	Big	Y	N	Decor
29	3	Mango Jam	Yellow	Big	Y	N	Food
33	3	Honey Jar	Yellow	Medium	Y	N	Food
40	3	Mushroom	Yellow	Small	Y	N	Food
56	3	Pumpkin	Yellow	Big	Y	N	Decor
60	3	Mango Jam	Yellow	Big	Y	N	Food

```
SELECT * FROM items WHERE is_edible = 'Y' AND Color = 'Yellow';
```

The query retrieves all the edible items from the items table where the item color is Yellow.

Methods Used:

SELECT – To retrieve the data from the Database.

FROM – Identifies the tables from which the data should be retrieved.

WHERE – Used to filter out the data based on the given condition.

AND – Used to filter records based on more than one condition (both conditions should be satisfied)

Thank you!