

CHAPTER - 8

ACCOUNTING SYSTEM USING DATABASE MANAGEMENT SYSTEM

Q1 :

State what you understand by accounting reports.

Answer :

The accounting reports are the physical form of accounting information. They act as a summary of all the relevant facts and information related to a particular event. The accounting reports are basically the end (or final) reports depicting the performance of an organisation during an accounting period. They show the information that is acquired from processing and transforming the data in an organised manner. The accounting reports display the information content in such a manner that the users can understand them without any ambiguity. These reports form the basis of the decision making process as they minimise the risks associated with uncertainty. Ledgers, Trial Balance, Cash/Bank Book, Financial Statements etc. are some of the examples of accounting reports.

Q2 :

What do you mean by programmed or casual reports?

Answer :

Accounting reports can be classified into two broad categories- Programmed Reports and Casual Reports.

1. Programmed Reports- These are those reports that contain information, which can be utilised by the users in the situations that they expect to occur in the near future.

a) Scheduled Reports- These are the reports that are prepared for a specific period of time. These reports can be prepared regularly on daily, weekly, quarterly, monthly or yearly basis. Trial Balance, Ledgers, Statements of Cash Transactions, etc. are some of the examples of Scheduled Reports.

b) On-Demand Reports- These are the reports that are prepared as and when required or on the happening of some particular event. Customer's Statement of Accounts, Inventory Re-order Report, Stock Purchased Report, etc. are some of the examples of On-Demand Reports.

2. Casual Reports- These are the reports that contain information which may be useful but are only casually required. The users do not anticipate their need. These are also known as ad hoc reports and can be generated randomly without much professional assistance.

Q3 :

With the help of an example, briefly state the meaning of parameter queries.

Answer :

A Parameter Query is a query where for selecting a set of records, the user is asked to enter parameters or criteria through an input box. A Parameter Query is useful when the same query is to be repeated with different criteria. In short, it implies the execution of same set of queries with different requirements.

For example, the same set of queries is required for different accounts code, while extracting the information to prepare ledger. Let us consider the given below SQL statement.

PARAMETERS Accounts Name Text (145)

SELECT Name

FROM Accounts

WHERE Code = AccountNo.

In the above query, PARAMETERS clause is used to represent the variable AccountNo. This SQL statement requires the user to provide the different values of Account No.

Q4 :

Briefly state the purpose of functions in SQL environment.

Answer :

In the access environment a function is named and followed by parenthesis (). The function takes some inputs as its argument and in turn sends a value. These functions act as a part of the expression for a computer field. In the SQL environment, following are the main types of functions that are used along with their respective purposes.

1. Domain Aggregate Functions- These functions are used to perform calculation that are based on the values in a field of a table or query. A domain aggregate function uses the syntax as- **DFunction** ("Fld Name", "Tbl Name" or Qry Name, "Srch Cond")

where,

DFunction represents Domain Aggregate Function

Fld Name represents Name of the field to be searched

Tbl Name or Qry Name represents Name of the table or query that contains the above field.

Srch Cond represents Search Condition with the help of which relevant record is searched.

2. SQL Aggregate Functions- These functions are used in those SQL statements that provide the underlying record source of forms and reports. All these functions when used in SQL statement require GROUP BY clause.

3. Other Functions- Apart from Domain Aggregate Function and SQL Aggregate Function, there are some other functions also that are used in SQL. These are mentioned below.

- a. **IIF-** These functions are used to provide a value to the field from a mutually exclusive set of values.
- b. **Abs-** These functions are used to provide the absolute value in return of a numeric value. It takes the absolute value as its input argument and gives back an absolute value.
- c. **Val-** These functions are used to provide numbers contained in a string as a numeric value of appropriate type.

Q5 :

Briefly explain in steps the method of creating a query, using wizard.

Answer :

The following are the steps involved in creating a query using the wizard method.

1. In the Database window, select 'Queries' from Object list given on the Left Hand Side (LHS) of the Database window.
2. Double click at 'Create Query' by using wizard given on the Right Hand Side (RHS) of the window. With this 'Simple Query Wizard' will be shown on the screen. Now, the user is required to select a field name from a table or from an existing query which should be included in the query being created.
3. Click 'Next' after the desired fields have been selected. In case, a number or currency field is included in the selected field, the user is asked to choose an option button to specify whether the query to be created is a summary or detail query.
4. Click 'Next' and specify the name of the query being created % 'Finish' to save and execute the query. The results of the query created are displayed in the datasheet view.

Q6 :

List the structure of a good report created in Access.

Answer :

A good report designed in the access must have the following sections -

1. **Report Header-** It appears at the top of the report and includes title and other relevant information related to the report.
2. **Page Header-** It appears at the top of every page of the report and includes a uniform title to indicate that the page belongs to a particular report.
3. **Group Header-** It appears only when the report contains the sort order and grouping levels. The sort order and the grouping headers are defined on the basis of a field of data source. Group headers appear at the top of each report group. There are separate group headers for each group level.
4. **Details-** It is also regarded as the main body of a report. It contains data from tables or queries that provide the record source to a report. This section consists of main information content of a report, hence is considered to be the most important section.
5. **Group Footer-** It appears at the bottom of each grouping level and contains summaries or sub-totals for the grouped data.
6. **Page Footer-** It also appears at the bottom of each page of the report and shows page numbers, date and time of report generation.
7. **Report Footer-** It appears on the last page of the report and includes summaries or totals for all the data of the report.

Q7 :

List the ways to refine the design of a report.

Answer :

A report design can be refined using the following ways.

1. **Adding Dates and Page Numbers-** In the design mode, the page footer of a report contains two unbound controls- 'current date' and 'current page number of the total number of pages'. The current date can be retrieved from the RTC using 'Now ()' function. Similarly, the date and time can also be added by clicking 'Insert % date and time' from the menu bar to open the Date and Time dialog box.
2. **Adding and Deleting Report Controls-** After designing the report, one can add or delete additional report controls by the same procedures as of forms.
3. **Conditionally Formatting the Report Controls-** With the help of Conditional Formatting, the designer can apply special text formats that depend on the value of a field. Such a formatting helps to draw the attention of the users or readers of the report to some values of particular interest.
4. **Grouping Levels and Sorting Order-** The Grouping Levels on one hand organises the information content of a report into various categories, while on the other hand, the Sorting Order organises the information content into numerical or alphabetical order. The applications of both the orders make the report more meaningful, thereby more useful to the users.

Q8 :

Briefly explain the purpose of grouping and sorting of the data as a means to refine a report.

Answer :

The purpose of grouping is to arrange the information content of a report into various categories, while on the other hand, the main purpose of the sorting order is to organise the information content into numerical or alphabetical order. The applications of both the orders make the report more meaningful, thereby more useful to the users.

Q9 :

What do you understand by saving a report as snapshot?

Answer :

Saving a report as a snapshot file enables the users to view it without the help of access. When a report is saved as a snapshot, a high quality picture image of each page of the report is created with Adobe Acrobat Software. With this the users can view the report and print any of its pages. However; the users cannot modify the contents of the reports in a snapshot file.

Q10 :

State the procedure for creating ledger in MS Access.

Answer :

To produce a ledger, it is essential to retrieve a set of processed data records providing information on Code (Account Number), Name of Accounts, Particulars, Date, Debit Balance and Credit Balance with reference to each particular account.

In order to get all this information, it is necessary to make certain SQL statements and save them as Query. Suppose, we have saved the last query as 'query L'. In Access, a ledger account can be created by taking the following steps in the design view method.

1. Select 'Reports' from the 'Object' list in the Database window.
2. Click 'New' button so that 'New Report' window is displayed.
3. Select 'Design View' and 'Query L' from the combo control.
4. Click 'OK'.

With this, the access responds by displaying a blank report design divided horizontally into three sections- Page Header, Detail and Page footer. Also, a list of available fields of Query L is provided for embedding on to this blank design.

5. Click at 'Properties' of the report and select 'Data' tab to define the recorded source as Query L. This displays a list of available fields of Query L.
6. Right click at any part of the report design and choose Report, Page Header and Footer. With this, the Access responds by providing two more sections- Page Header and Pager Footer.
7. Click at the 'icon' of the toolbar and pick up a label control to be placed at the Page Header section and assign its caption.
8. Select all the fields of Query L by clicking at every field, while keeping the 'Ctrl key' pressed. Drag and drop the selected field on the 'Details Section'.
9. Select the total control of all the six fields by clicking at each, while keeping the 'Shift key' pressed. Right click at the 'Select Label Control' and choose 'Cut'. Now, place the mouse at the Page Header Section and 'Paste' these controls.
10. Choose the 'Properties' provided by the Access.
11. Align the text controls in the 'Details Section'.
12. Select the 'Text Controls' and 'Amount Field' and modify their properties. Hence, the required ledger in access is created.

Long answers : Solutions of Questions on Page Number : 617

Q1 :

Describe and discuss the procedure of creating the receipts side of a cash book.

Answer :

The procedure of creating the receipts side of cash book is similar to report generation.

To produce the receipts side of the Cash Book, it is essential to retrieve a set of processed data records providing information on Code (Account Name), Name of Accounts, Particulars, Date Debit Balance and Credit Balance with reference to each particular account, where cash is debited.

In order to get all this information, it is necessary to make certain SQL statements and save them as Query in the Database.

In Access, the receipts side of a Cash Book can be created by following the below mentioned steps.

1. Select the 'Create Report' in the design view. With this, the access responds by displaying a blank report design divided into three sections- Page Header, Detail and Page Footer of the available fields of the Query.
2. Now, we design our report with the tools provided by the Access such as, naming the report, defining its size, colour, etc.
3. Select all the fields from the Query, drag and drop them into the 'Details Sections'. Hence, the receipts side of the Cash Book is created.

Q2 :

Discuss the concept of accounting reports? Explain the three steps involved in creating such reports.

Answer :

The accounting reports are the physical form of accounting information. They act as a summary of all the relevant facts and information related to a particular event. The accounting reports are basically the end (or final) reports depicting the performance of an organisation during an accounting period. They show the information that is acquired from processing and transforming the data in an organised manner. The accounting reports display the information content in such a manner that the users can understand them without any ambiguity. These reports form the basis of the decision making process as they minimise the risks associated with uncertainty. Ledgers, Trial Balance, Cash/Bank Book, Financial Statements etc. are some of the examples of accounting reports. An accounting report can be useful, if it has five basic characteristics, namely, Relevance, Timeliness, Accuracy, Completeness and Summarisation.

The accounting reports can be classified into two following broad categories.

1. Programmed Reports- These are those reports that contain information, which can be utilised by the users in the situations that they expect to occur in the near future.

a) Scheduled Reports- These are the reports that are prepared for a specific period of time. These reports can be prepared regularly on daily, weekly, quarterly, monthly or yearly basis. Trial Balance, Ledgers, Statements of Cash Transactions, etc. are some of the examples of Scheduled Reports.

b) On-Demand Reports- These are the reports that are prepared as and when required or on the happening of some particular event. Customer's Statement of Accounts, Inventory Re-order Report, Stock Purchased Report, etc. are some of the examples of On-Demand Reports.

2. Casual Reports- These are the reports that contain information which may be useful but are only casually required. The users do not anticipate their need. These are also known as ad hoc reports and can be generated randomly without much professional assistance.

Process of Creating Accounting Reports

In Access, the generation of accounting reports involves the following three steps.

Step 1: Designing the Report

A report must be designed in a manner, so as to meet certain objectives of reporting. It should not be either too long as not to be read or too short so as to conceal certain important information that is expected to help in the decision making process.

Step 2: Identification of Accounting Information Queries

Many SQL statements are written in such a manner that each successive SQL depends on the results of the preceding SQL statement. It then, refines its results by using fresh data from the existing data tables.

Step 3: Using the Record Set of Final SQL

The record set of final SQL is a collection of report-oriented information. It relies upon the preceding SQL statement. This record set must be included in the report being produced.

Q3 :

Discuss with a set of inter-related data tables, the basics of creating queries in MS Access?

Answer :

A query is a statement that gives refined data according to the conditions and specifications of the user. It can retrieve the desired data from the multiple data simultaneously and can also update or delete multiple records at the same time. It is a link between the interrelated tables, forms, etc.

Accounting information that is presented in an accounting report is generated by creating and executing various queries under DBMS. A query communicates to Access the kind of information which we require to extract from one or more interrelated tables.

The following are the basics of creating queries in the MS-Access with a set of inter-related data tables.

As Relational DBMS stores data in different tables, so there is no or minimum data redundancy. But a complete view of data that is stored across various tables can be achieved only by executing queries based on SQL. A query shows the records containing fields from across a number of data tables. A SQL statement has a number of parameters such as data definition, query, update, capability to define user-oriented views of database, specify security and authorisation and defining integrity constraints.

Example:

The procedure of creating queries for inter-related data tables using SQL statements is explained below by the help of an example.

SELECT Code, Name, Sum (Amount)

FROM Vouchers INNER JOIN Accounts.

ON (Accounts, Code = Voucher Debit)

GROUP BY Code, Name.

In this query, the 'Vouchers' table has been joined with 'Accounts' table on the basis of 'Code' fields of 'Accounts' and 'Debit' field of vouchers. The result record set has been grouped on the basis of Code and Name of Accounts.

Q4 :

Briefly explain the set of SQL statements to produce the receipts side of a cash book for Model-I.

Answer :

The following are the set of SQL statements that are used to produce the receipts side of a Cash Book for Model-I.

Step1: To Ascertain the Total Amount by which the Cash Account is Debited

In order to find out the total amount by which Cash Account has been debited in all the accounts the SELECT clause is executed. This clause has two following fields.

(a) A code to identify the account

(b) A code to generate the total by which the account has been debited

SQL Statement

SELECT Credit As Code, Amount, Date

FROM Vouchers

WHERE Debit Like "231"

This SQL statement is saved as query 01.

This statement provides the code and amount of cash received.

Step2: To Generate the Total of Receipts Side

SQL Statement

SELECT Code **Sum** (Amount) **As** Total

FROM Query 01

GROUP BY Code

This SQL statement is saved as Query 02

Step3: To Generate the Record Set Consisting of Account Code, Name of Account, Credit Amount and Date

SQL Statement

SELECT a.code b.name **As** [Name of Account], **IIF** (a = Total > 0, a. Total, null as Amount)

FROM Query 02 **As** a Accounts **As** b

WHERE a code=b code

Q5 :

Describe in steps the design view method to create a query in MS Access?

Answer :

To create a query using design view method, the following steps are followed

Step1: Select 'Queries' from the object list given on the left hand side of the database window.

Double click at 'Create Query by using Design View' given on the right hand side.

Step 2: With this, a 'Select Query' and 'Show Tables' window is displayed. The 'Select Query' window is divided into two panes namely, the upper pane (displays the data sources) and the lower pane also known as Query by Example (QBE) (displays column and row). The column of the lower pane displays the data sources. The row of the lower pane displays the field name, table (or query), sort order, whether the selected field is shown in the query results or not and also the criteria that have been applied to the fields to restrict the query results.

Step 3: Click at the 'View' item of the Menu Bar % Total and then % Table Names.

Step 4: Click at the 'Field Row' of the first column of the QBE and select the fields to be included in the query. The process is repeated for the subsequent columns of the QBE to include more fields in the query. This process of selection contains the data items that are to be displayed by the SELECT clause of the SQL statement.

Step 5: Now, in accordance to the selection of the fields, the name of the table or query is displayed. Such tables or queries constitute the data sources shown after the FROM clause of the SQL statement.

Step 6: Click at 'Row of the low pane' to specify the group by clause and aggregate functions to create a summary of the query is created.

Step 7: Click at 'Row of the low pane' to specify the sort order (Ascending or descending). The selected fields for sort order are shown after ORDER BY clause of the SQL statement in which ascending order is a default choice.

Step 8: Click at row to display the selected field in the query result.

a) Click at row of the grid to specify the criteria to limit the records to be displayed by the query being created. The specified criteria result is shown after the WHERE clause of the SQL statement.

b) Click File % Save (or Press Ctrl + S) to save the query. Now, a dialog box appears that asks the user to specify the name of the query being created.

Q6 :

Discuss the SQL view method of creating a query?

Answer :

The following clauses are used for generating information queries using SQL statements.

1. SELECT: This clause specifies the fields to display data or information. For example SELECT Code, Name, Amount

This statement indicates that the fields Code, Name and Amount are to be displayed by the query statement.

2. FROM: This clause specifies the source of data in terms of tables or queries or a combination of both. By specifying the JOIN clause, two tables are joined. Join clause can be of following three types.

a) **INNER-** This Join clause displays only those records that are exactly matching between two data sources. For example, FROM Accounts INNER JOIN AccountType ON (CatId = Type)

Here, only those records of Accounts and AccountType table are a part of source of query data, which match exactly on CatId = Type.

b) **LEFT-** This JOIN clause displays all the records in the primary table in a relationship irrespective of whether there are matching records in the related table or not. For example, FROM Accounts LEFT JOIN AccountType ON (CatId = Type)

Here, all the records of Accounts along with the matching records form a part of the source of the query data.

c) **RIGHT-** This JOIN clause displays all the records of related table in a relationship irrespective of whether there are matching records in the primary table or not.

For example, FROM Accounts RIGHT JOIN AccountType ON (CatId = Type)

Here, all records of the AccountType along with the matching records of the Accounts table form a part of the source of query data.

3. WHERE: This clause provides the condition to restrict the records to be returned by the query. The result records of the query must satisfy the condition which is specified after the WHERE clause.

4. ORDER BY- This clause specifies the order in which the resultant records of the query are required to appear. The basis on which this ordering is done is determined by the list of the fields specified after the ORDER BY clause. For example, ORDER BY Type, Code

This statement implies that the resultant record set is ordered by the 'Type' field of Accounts and within Type, by 'Code' field of Accounts.

5. GROUP BY- This clause enables the grouping of records for creating summary query. The fields specified after the GROUP BY clause form the basis of grouping for which the summary results are obtained. For example,

SELECT Debit, Sum (Amount)

FROM Vouchers

GROUP BY Debit

Here, the GROUP BY clause uses debit account codes for calculating the sum total of the amount of the vouchers.

Q7 :

Describe the ways to refine the design of a report.

Answer :

A report design can be refined using the following ways-

1. Adding Dates and Page Numbers- In the design mode, the page footer of a report contains two unbound controls- 'current date' and 'current page number of the total number of pages'. The current date can be retrieved from the RTC using 'Now ()' function. The page number can be added by clicking 'Insert % page' numbers from the menu bar to open the page numbers dialogue box. With this, the designer can specify the format, position and alignment. Similarly, the date and time can also be added by clicking 'Insert % date and time' from the menu bar to open the Date and Time dialog box.

2. Adding and Deleting Report Controls- After designing the report, one can add or delete additional report controls by the same procedures as of forms.

- a. Open the report in the design view and click 'Field List' button on report design tool bar.
- b. Now, drag the field into an appropriate section of the report, so that the field appears with both label and text box control.
- c. One can also delete a field by selecting the control and pressing the 'Delete' key.

3. Conditionally Formatting the Report Controls- With the help of Conditional Formatting, the designer can apply special text formats that depend on the value of a field. Such a formatting helps to draw the attention of the users or readers of the report to some values of particular interest.

The following steps are followed to create a Conditional Formatting.

- a. Open report in the design view and select a control and click at 'Format' on the menu bar.
- b. Now, click at 'Conditional Formatting' and provide the necessary conditions for formatting.
- c. Conditional Formatting can also be deleted by reopening the same dialog box and clicking the 'Delete' button.

4. Grouping and Sorting of Data- The Grouping Levels on one hand organises the information content of a report into various categories, while on the other hand, the Sorting Order organises the information content into numerical or alphabetical order. The applications of both the orders make the report more meaningful, thereby more useful to the users.

Q8 :

Explain the data base design for Model-I for producing the receipts the series of SQL statements for producing the payment side of cash book for Model-II.

Answer :

The following series of SQL Statement retrieve a record set of producing the payments side of the Cash Book for Model II.

(1) **SELECT** Narration, Acc-code **As** Code Amount

FROM vouchers **As** v, Details **As** D

WHERE tType=1 **AND** V.VNO=D.VNO

AND Acc-code like "141*".

(2) **SELECT** Narration, Code, Amount

FROM vouchers **As** v, Details **As** D

WHERE tType = 0 **AND** V.VNO = D.VNO

AND Code **Like** "141*".

(3) **SELECT** Narration, Acc-code **As** Code Amount

FROM vouchers **As** v, Details **As** D

WHERE tType=1 **AND** V.VNO=D.VNo

AND Acc-code **Like** "141*".

UNION

SELECT Narration, Code, Amount

FROM vouchers **As** v, Details **As** D

WHERE tType=0 **AND** V. VNO=D.VNo

AND Acc-code like "141*".

This SQL statement is source as Query D1

(4) **SELECT** Code, Sum (Amount) **As** Total

FROM Query D1

GROUP BY Code

This SQL statement is saved as Query D2.

(5) **SELECT** a.Code b.Name **As** [Name of Account],

IIF (a.total>0, a total **Null**) **As** Amount

FROM Query D2 **As** a, Account **As** b

WHERE a code=b code

Q9 :

Describe the series of SQL statements to produce trial balance data base design for Model-II is used.

Answer :

In the following set of SQL statement, the accounts have been categorised within the Trial Balance according to the Account Type as: Expenses, Revenues, Assets and Liabilities.

The below mentioned series of SQL Statement retrieve the record set for producing Trial Balance when database design for Model-II is used.

SQL Statements

1. To ascertain the total amount by which the accounts have been debited

The transacted accounts have been stored AccCode of vouchers main and code of Vouchers Details.

SQL Statement for Vouchers Details

SELECT Code, **SUM** (amount), **As** total

FROM vouchers Main **INNER JOIN** voucher Details

ON Voucher Main. VNO=voucher Details. VNO

WHERE Type=0

GROUP BY Code;

SQL Statement for Vouchers Main Table

SELECT AccCode **As** Code, sum (amount), **As** total

FROM vouchers Main **INNER JOIN** voucher Details **ON**

Voucher Main. VNO=vouchers Details. VNO

WHERE Type=1

GROUP BY AccCode;

Now, we need to join the above mentioned SQL statements using UNION clause. This is done by the SQL command.

SQL Statement for Merging Voucher Details and Vouchers Main Table

SELECT Code, sum (amount), **As** total

FROM vouchers Main **INNER JOIN** voucher Details **ON**

Voucher Main. VNO=voucher Details. VNO

WHERE Type=0

GROUP BY Code;

UNION ALL

SELECT AccCode **As** Code, sum (amount), **As** total

FROM vouchers Main **INNER JOIN** voucher Details **ON**

Voucher Main. VNO=voucher Details. VNO

WHERE Type=1

GROUP BY AccCode;

This SQL Statement is saved on Query 01 for its subsequent use.

The total of debit amount in this query represents a positive total.

2. To ascertain the total amount by which the accounts have been credited

The following series of SQL statements help in ascertaining the total by which every transacted account has been credited.

SELECT Code, sum (amount)* -1, **As** total

FROM vouchers Main **INNER JOIN** voucher Details **ON**

Voucher Main. VNO=voucher Details. VNO

WHERE Type=1 **GROUP BY** Code, Amount

UNION

SELECT AccCode As Code, sum (amount)* -1, **As** total

FROM vouchers Main **INNER JOIN** voucher Details **ON**

Voucher Main. VNO=vouchers Details. VNO

WHERE Type=0 **GROUP BY** AccCode, Amount;

We can see that the total of the credit amount has been multiplied by (-) 1. This is to ensure that the total credit amount should be in negative on contrary to the positive total of debit.

This query is saved as Query 02.

3. To compute a collective record set of accounts with their debit and credit totals

In order to pool out a collective record set of accounts, a union query between the saved Query 01 and Query 02 is passed. This is done to ensure that the debit and the credit amount with respect to each account become available for generating the net amount (Debit - Credit).

The required SQL Statement for Union query

SELECT*

FROM Query 01

UNION Select*

FROM Query 02

This statement results in a horizontal merger of the above queries, viz. Query 01 and Query 02. This SQL statement is saved as Query 03.

4. To ascertain the net amount with which an account has been debited or credited

SQL Statement

SELECT Code, **Sum** (total), **As** Net

FROM Query 03

UNION Select*

GROUP BY Code;

This SQL statement is saved as Query 04.

5. To form the record set which consists of Account Code, Name of Account, Debit Amount and Credit Amount

This query provides relevant information to the Trial Balance report.

SQL Statement

```

SELECT a. Code b. name As [Name of Account],
IIF (a. Net>0, null) As Debit, IIF (a. Net<0, abs(a.Net), null)
As Credit FROM Query 04 As a, Account As b/
WHERE a.code=b.code;

```

This SQL statement is saved as Query 05

Trial Balance with sorting and Grouping levels

6. To ascertain the record set of accounts with their category and category ID

SQL Statement

```

SELECT Accounts.Code, Accounts Name, Category, CatId
FROM Accounts
INNER JOIN Account Type ON

```

Accounts. Type=Account type, CatId;

This SQL statement is saved as Query 06.

7. To ascertain the record set consisting of Account Code, Name of Account, Debit Amount and Credit Amount along with their category details

SQL Statement

```

SELECT a.Code, b.name As [Name of Account];
IIF (a. Net>0, a. Net, null) As Debit, IIF (a. Net<0,abs(a.Net), null) As Credit, Category, CatId
FROM Query 04 As a, Query 06 As b
WHERE a.code=b.code;

```

This SQL statement is saved as Query 07. This query provides information details for designing Trial Balance with grouping and sorting of the accounts.

Q10 :

Using Model-I discuss the series of SQL statements to produce a trial balance up to a particular date.

Answer :

The below mentioned series of SQL Statement retrieve the record set for producing Trial Balance when database design for Model-I is used.

1. To ascertain the total amount by which the accounts have been debited

SQL Statement

```

SELECT Debit As Code, Sum (amount), As total
FROM vouchers
GROUP BY Debit;

```

This SQL statement is saved as Query 01.

The GROUP BY clause retrieves the rows of vouchers table accounts-wise. The total of the debit amount in this query is given by the Total field with positive amounts.

2. To ascertain the total amount by which the accounts have been credited

SQL Statement

```

SELECT Credit As Code, Sum (amount)* (-1) As total

```

FROM vouchers

GROUP BY Credit;

This SQL statement is saved as Query 02.

We can see that the total of the credit amount has been multiplied by (-) 1. This is to ensure that the total credit amount should be in negative on contrary to the positive total of debit.

3. To generate a collective record set of accounts with their debit and credit totals

This collective record set is generated by executing a union query between the above queries viz. Query 01 and Query 02.

SQL Statement

SELECT*

FROM Query 01

UNION SELECT*

FROM Query 02

This SQL statement is saved as Query 03.

4. To ascertain the net amount with which an account has been debited or credited

SQL Statement

SELECT Code, **Sum** (total), **As** Net

FROM Query 03

GROUP BY Code

This SQL statement is saved as Query 04.

A positive net amount implies a debit and negative amount means a credit balance corresponding to an account code. This query can be used for generating record set for Trial Balance.

5. To generate the record set which consists of account Code, name of account, debit amount and credit amount

SQL Statement

SELECT a.Code, b.Name **As** [Name of Account]

IIF (a.Net>0,a.Net,**null**) **As** Debit,

IIF (a.Net<0,abs (a.Net), **null**) **As** Credit,

FROM Query 04 **As** a, Account **As** b

WHERE a.code=b.code;

This SQL statement is saved as Query 05. This query can be used for providing the necessary information content for the Trial Balance Report.

Important Note: *There is a mistake in the Question No. 10 of the book. The correct question should be on Model- I instead of Model-III.*