

***A signed copy of this form must be submitted with every assignment.
If the statement is missing your work may not be marked.
For Level 7 assignments, please use the separate Level 7 Candidate
Statement of Own Work form instead of this one.***

Student Declaration

I confirm the following details:


Candidate Name:	ISHAN BAPARDEKAR
Candidate ID Number:	213062
Qualification:	L4DC
Unit:	DBMS
Centre:	FUT004
Word Count:	1071
<p>I have read and understood both NCC Education's <i>Academic Misconduct Policy</i> and the <i>Referencing and Bibliographies</i> document. To the best of my knowledge my work has been accurately referenced and all sources cited correctly.</p> <p>I confirm that I have not exceeded the stipulated word limit by more than 10%.</p> <p>I confirm that this is my own work and that I have not colluded or plagiarised any part of it.</p>	
Candidate Signature:	
Date:	1503

Table of Contents

Task 1 creating ER diagram	1
a) ER diagram	2
b) Justification and Explanation for the inclusion of Entities	3
Task 2: Data dictionary	4
TASK 3 Database Creation:	5
a) SQL commands for creation and insertion of values:	6
Task 4 – 15 Marks Role-based Access Control	7
a) Implement role-based access mechanisms	8
Task 5 SQL queries	9
a) Write a SQL query to update the inventory to reflect the latest pricing after a supplier price change. This query increases the price of all gardening tools by 10% to adjust for an increase in supplier costs.	10
b) Write a SQL query to retrieve the highest priced item from each of the following categories: plants, gardening tools and accessories. Provide the product name, type, price and a brief description for each.	11
c) Write a SQL query to summarise the total sales transactions for a specific past month, breaking down the data by item category, including quantities sold and total revenue per category.	12
d) Write a SQL query to find all upcoming workshops and events within the next month, including the titles, dates, times, and the current number of registered participants.	13
e) Write a SQL query to retrieves and rank employees based on the total revenue they've generated from these transactions up until a specific date.	14

TASK 1 - Create an ER diagram:

a) ER Diagram:

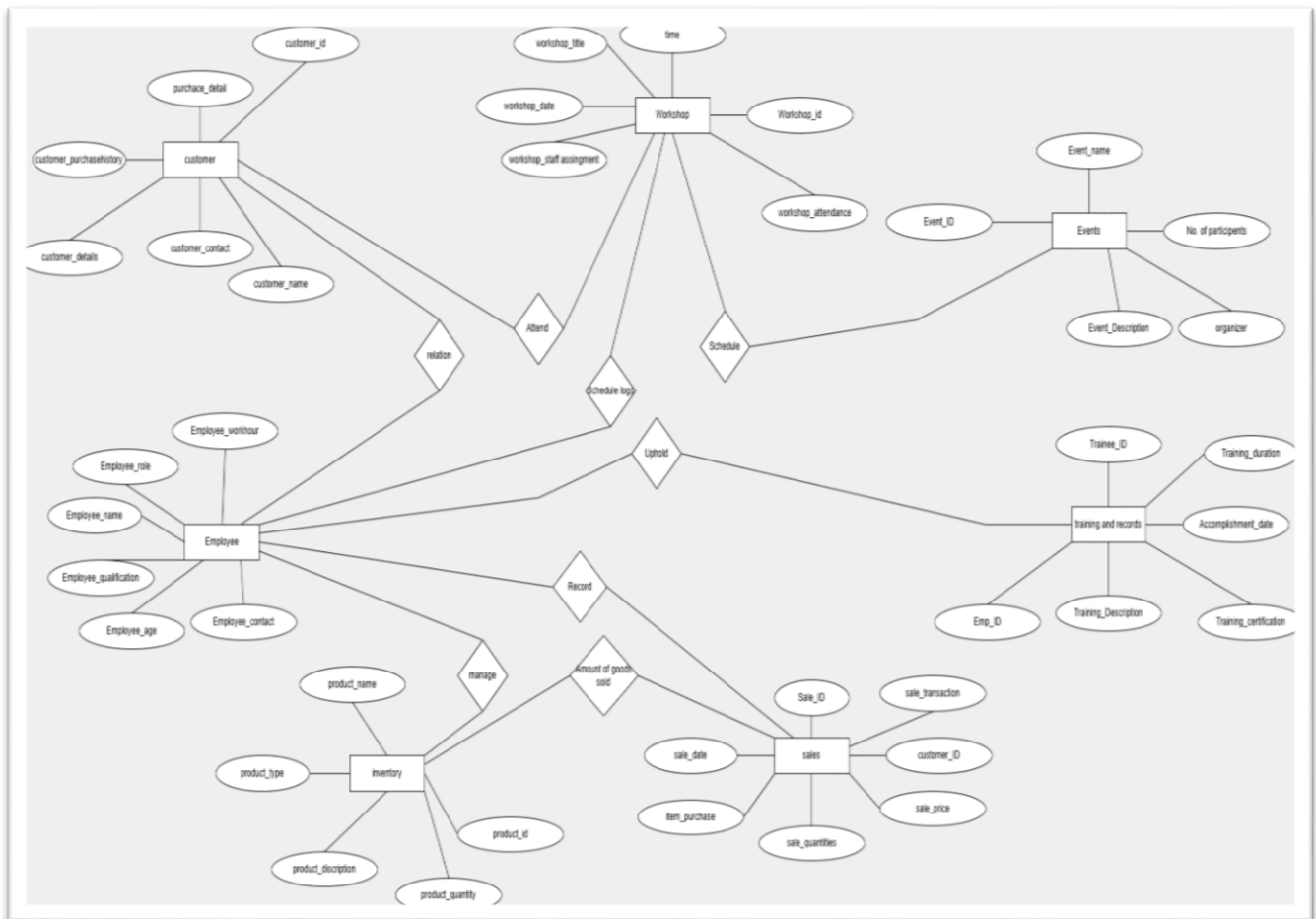


Fig 1.1 ER Diagram

b) Justification and Explanation for the inclusion of Entities:

Entity	Justification
Customer	Captures essential customer information including name and contact. Required to link purchases, workshop attendance, and sales.
Customer_PurchaseHistory	Maintains historical purchase information of a customer. Supports customer behavior analysis.
Employee	Stores staff information necessary for workshop management, training records, and administrative functions.
Inventory	Holds product-related data for sale and stock tracking. Needed to manage quantity and types of items available.
Sales	Tracks individual sales transactions, linking customers with purchased items, prices, and dates.
Workshop	Represents learning or training sessions offered by the business. Includes schedule, staff, and attendance.
Events	Records event details like name, number of participants, and description. Useful for marketing and internal planning.
Training_and_Records	Maintains training data of employees including certification and duration. Supports HR and performance tracking.

Table 1.1: entity and justification table

Normal Form	Explanation
First Normal Form	All attributes in each entity are atomic (indivisible). Repeating groups are removed. For eg. in Customer, customer contact is single-valued, and no multi-valued fields exist.
Second Normal Form	All attributes are fully functionally dependent on the primary key. Partial dependencies are removed. For eg. in Sales, sale_price, sale_date, and item_purchase all depend entirely on sale_transaction.
Third Normal Form	There are no transitive dependencies — non-key attributes do not depend on other non-key attributes. For eg. in Employee, attributes like Employee_age, Employee_qualification directly depend on Emp_ID.

Table 1.2 normal form and explanation

Task 2: Data dictionary

1. Customer

Attribute	Data Type	Field Length	Constraints
customer_id	INT	10	Primary Key, Not Null, Unique
customer_name	VARCHAR	100	Not Null
customer_contact	VARCHAR	50	Nullable

Table 2.1: customer

- Customer_ID uniquely identifies customer
- Customer contact helps contacting customer

2. Customer_PurchaseHistory

Attribute	Data Type	Field Length	Constraints
purchase_detail	TEXT	-	Nullable
customer_id	INT	10	Foreign Key (Customer.customer_id)

Table 2.2: customer purchase history

- Purchase_detail helps to keep track of customer purchases

3. Employee

Attribute	Data Type	Field Length	Constraints
Employee_id	INT	10	Primary Key, Not Null, Unique
Employee_name	VARCHAR	100	Not Null
Employee_contact	VARCHAR	50	Nullable
Employee_title	VARCHAR	50	Nullable
Employee_qualification	VARCHAR	100	Nullable
Employee_workhour	INT	3	Default 0, Check (≥ 0)

Table 2.3: employee

- Employee_ID uniquely identifies employees
- Employee workhour makes it easy to pay the employee based on workhour
- Employee_qualification employee is expected to work according to their qualifications

4. Inventory

Attribute	Data Type	Field Length	Constraints
product_id	INT	10	Primary Key, Not Null, Unique
product_name	VARCHAR	100	Not Null
product_type	VARCHAR	50	Nullable
product_description	TEXT	-	Nullable
product_quantity	INT	5	Default 0, Check (≥ 0)

Table 2.4: inventory

- Product ID uniquely identifies product

5. Sales

Attribute	Data Type	Field Length	Constraints
Sale_ID	INT	10	Primary Key, Not Null, Unique
sale_transaction	INT	10	Not Null, Unique
customer_ID	INT	10	Foreign Key → Customer.customer_id, Not Null
item_purchase	VARCHAR	100	Not Null
sale_price	DECIMAL	10,2	Not Null, Check (≥ 0)
sale_quantities	INT	5	Not Null, Check (≥ 0)
sale_date	DATE	-	Not Null

Table 2.5 sales

- Sale_ID uniquely identifies sales

6. Workshop

Attribute	Data Type	Field Length	Constraints
Workshop_ID	INT	10	Primary Key, Not Null, Unique
workshop_title	VARCHAR	100	Not Null
workshop_date	DATE	-	Not Null
time	TIME	-	Nullable
workshop_attendance	INT	5	Default 0, Check (≥ 0)
workshop_staff_assignment	VARCHAR	100	Nullable

Table 2.6: workshop

7. Events

Attribute	Data Type	Field Length	Constraints
Event_ID	INT	10	Primary Key, Not Null, Unique
Event_name	VARCHAR	100	Not Null
No_of_participants	INT	5	Default 0, Check (≥ 0)
Event_Description	TEXT	-	Nullable
organizer	VARCHAR	100	Nullable

Table 2.7: events

8. Training and Records

Attribute	Data Type	Field Length	Constraints
Trainee_ID	INT	10	Primary Key or Unique, Not Null
Emp_ID	INT	10	Foreign Key (Employee.Employee_id)
Training_Description	TEXT	-	Nullable
Training_duration	INT	3	Check (≥ 0), Nullable
Accomplishment_date	DATE	-	Nullable
Training_certification	VARCHAR	100	Nullable

Table 2.8: trainings and records

9. Relationships

Relationship	Entities Involved	Description
Attend	Customer - Workshop	Tracks which customer attends which workshop
Record	Sales - Inventory	Records which products are sold
Manage	Employee - Inventory	Tracks which employee manages inventory
Schedule for	Employee - Workshop	Assigns employee schedules for workshops
Uphold	Employee - Events	Maintains responsibility for organizing
Relation	Customer - Customer_PurchaseHistory	Links customer to their purchase records

Table 2.9 relationships

TASK 3 Database Creation:

a) SQL commands for creation and insertion of values:

1. Create data base :

```
1 • CREATE DATABASE BLOOM_FIELD_GARDEN_CENTRE;
2 • USE BLOOM_FIELD_GARDEN_CENTRE;
```

Fig 3.1: SQL for Database Creation

2. Customer table

Creation:

```
1 CREATE TABLE customer (
2     CustomerID INT PRIMARY KEY,
3     CustomerName VARCHAR(100),
4     CustomerContact VARCHAR(100),
5     CustomerDetail TEXT,
6     PurchaseDetail TEXT,
7     CustomerPurchaseHistory TEXT
8 );
9 • DESCRIBE customer;
10
```

Field	Type	Null	Key	Default	Extra
CustomerID	int	NO	PRI	NULL	
CustomerName	varchar(100)	YES		NULL	
CustomerContact	varchar(100)	YES		NULL	
CustomerDetail	text	YES		NULL	
PurchaseDetail	text	YES		NULL	
CustomerPurchaseHistory	text	YES		NULL	

Figure 3.2: SQL for Creating customer Table

Insertion:

```

1 SELECT * FROM DATABASE.garden_centre.customer;
2 INSERT INTO Customer ( CustomerID, CustomerName, CustomerContact, CustomerDetail, PurchaseDetail, CustomerPurchaseHistory
3 ) VALUES
4 (1, 'Emma Green', 'emma.green@example.com | 07890123456',
5  'Frequent workshop attendee. Prefers organic gardening.',
6  'Purchased: 2x Lavender Plants, 1x Watering Can (£25.97)',
7  'Mar 12, 2025: Lavender Plants, Watering Can | Jan 10, 2025: Rosemary, Garden Gloves'),
8
9 (2, 'James Taylor', 'james.taylor@example.com | 07987654321',
10 'Interested in eco-friendly tools. Attended composting workshop.',
11 'Purchased: 1x Compost Bin (£45.00)',
12 'Apr 5, 2025: Compost Bin | Feb 18, 2025: Organic Soil, Planter Pots'),
13
14 (3, 'Sophia Khan', 'sophia.khan@example.com | 07700112233',
15 'Regular buyer of seasonal flowers. Brings family to events.',
16

```

CustomerID	CustomerName	CustomerContact	CustomerDetail	PurchaseDetail	CustomerPurchaseHistory
1	Emma Green	emma.green@example.com 078901...	Frequent workshop attendee. P...	Purchased: 2x Lavender Plants, 1x Water...	Mar 12, 2025: Lavender Plants, Watering Can Jan 10, 2025: ...
2	James Taylor	james.taylor@example.com 079876...	Interested in eco-friendly tools...	Purchased: 1x Compost Bin (£45.00)	Apr 5, 2025: Compost Bin Feb 18, 2025: Organic Soil, Planter ...
3	Sophia Khan	sophia.khan@example.com 077001...	Regular buyer of seasonal flow...	Purchased: 3x Daffodil Bunches (£15.00)	Mar 30, 2025: Daffodils Dec 15, 2024: Poinsettias, Mulch
4	Liam Wong	liam.wong@example.com 07555666...	DIY enthusiast. Took part in rais...	Purchased: 1x Raised Bed Kit, 2x Tomato ...	Apr 10, 2025: Raised Bed Kit, Tomato Seeds Nov 5, 2024: Ha...
5	Olivia Patel	olivia.patel@example.com 0744433...	New customer. Signed up for he...	Purchased: 1x Basil Plant, 1x Fertilizer (£1...	Apr 15, 2025: Basil, Fertilizer

Fig 3.3: SQL for Inserting in customer Table

3. Employee table

Creation:

```

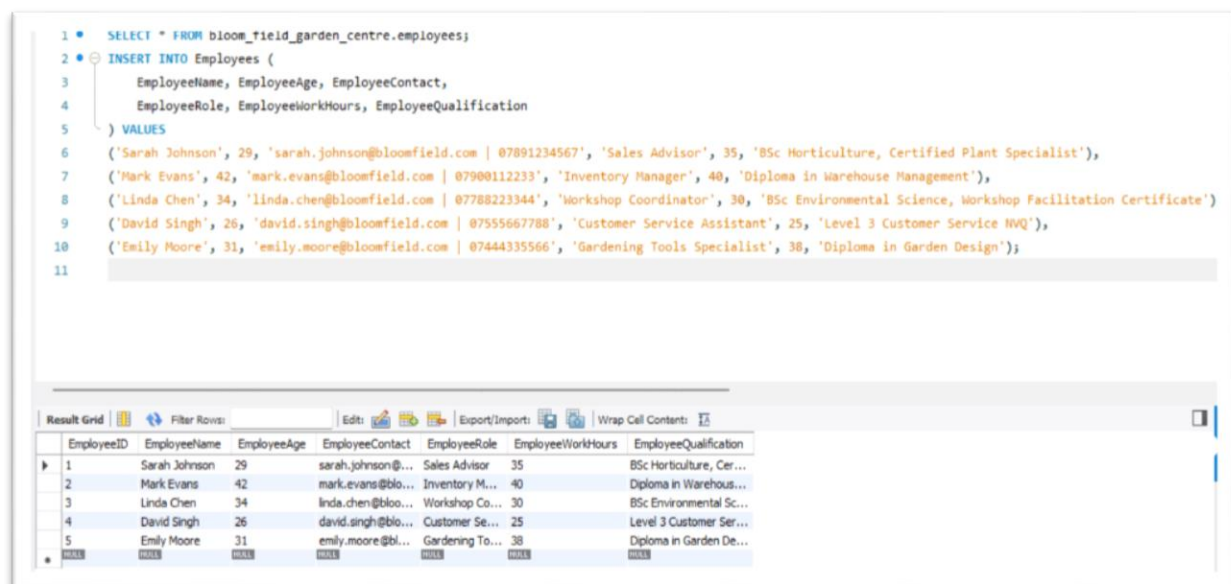
1 CREATE TABLE Employees (
2     EmployeeID INT AUTO_INCREMENT PRIMARY KEY,
3     EmployeeName VARCHAR(100) NOT NULL,
4     EmployeeAge INT,
5     EmployeeContact VARCHAR(100),
6     EmployeeRole VARCHAR(50),
7     EmployeeWorkHours INT,
8     EmployeeQualification TEXT
9 );
10 DESCRIBE Employees;

```

Field	Type	Null	Key	Default	Extra
EmployeeName	varchar(100)	NO		NULL	
EmployeeAge	int	YES		NULL	
EmployeeContact	varchar(100)	YES		NULL	
EmployeeRole	varchar(50)	YES		NULL	
EmployeeWorkHours	int	YES		NULL	
EmployeeQualification	text	YES		NULL	

Fig 3.4 SQL for Creating

Insertion:



The screenshot displays a SQL query in a text editor and its corresponding result grid. The query is an INSERT statement into the 'Employees' table of the 'bloom_field_garden_centre' database. It lists five new employees with their details: Sarah Johnson, Mark Evans, Linda Chen, David Singh, and Emily Moore. The result grid below the query shows the data that was inserted, with columns for EmployeeID, EmployeeName, EmployeeAge, EmployeeContact, EmployeeRole, EmployeeWorkHours, and EmployeeQualification.

```

1 • SELECT * FROM bloom_field_garden_centre.employees;
2 • INSERT INTO Employees (
3     EmployeeName, EmployeeAge, EmployeeContact,
4     EmployeeRole, EmployeeWorkHours, EmployeeQualification
5 ) VALUES
6 ('Sarah Johnson', 29, 'sarah.johnson@bloomfield.com | 07891234567', 'Sales Advisor', 35, 'BSc Horticulture, Certified Plant Specialist'),
7 ('Mark Evans', 42, 'mark.evans@bloomfield.com | 07900112233', 'Inventory Manager', 40, 'Diploma in Warehouse Management'),
8 ('Linda Chen', 34, 'linda.chen@bloomfield.com | 07788223344', 'Workshop Coordinator', 30, 'BSc Environmental Science, Workshop Facilitation Certificate'),
9 ('David Singh', 26, 'david.singh@bloomfield.com | 07555667788', 'Customer Service Assistant', 25, 'Level 3 Customer Service NVQ'),
10 ('Emily Moore', 31, 'emily.moore@bloomfield.com | 07444335566', 'Gardening Tools Specialist', 38, 'Diploma in Garden Design');
11

```

EmployeeID	EmployeeName	EmployeeAge	EmployeeContact	EmployeeRole	EmployeeWorkHours	EmployeeQualification
1	Sarah Johnson	29	sarah.johnson@...	Sales Advisor	35	BSc Horticulture, Cer...
2	Mark Evans	42	mark.evans@blo...	Inventory M...	40	Diploma in Warehous...
3	Linda Chen	34	linda.chen@blo...	Workshop Co...	30	BSc Environmental Sc...
4	David Singh	26	david.singh@blo...	Customer Se...	25	Level 3 Customer Ser...
5	Emily Moore	31	emily.moore@bl...	Gardening To...	38	Diploma in Garden De...

Fig 3.5 SQL for Inserting in employee

4. Inventory

creation:

```

1 CREATE TABLE Inventory (
2     ProductID INT AUTO_INCREMENT PRIMARY KEY,
3     ProductName VARCHAR(100) NOT NULL,
4     ProductQuantity INT NOT NULL,
5     ProductDescription TEXT,
6     ProductType VARCHAR(50)
7 );
8 DESCRIBE Inventory;
  
```

The screenshot shows a SQL editor with the above code. Below the editor is a 'Result Grid' displaying the table structure:

Field	Type	Null	Key	Default	Extra
ProductID	int	NO	PRI	NULL	auto_increment
ProductName	varchar(100)	NO		NULL	
ProductQuantity	int	NO		NULL	
ProductDescription	text	YES		NULL	
ProductType	varchar(50)	YES		NULL	

Fig.3.6 creating inventory table

Insertion:

```

2 INSERT INTO Inventory (
3     ProductName, ProductQuantity, ProductDescription, ProductType, ProductPrice
4 ) VALUES
5 ('Lavender Plant', 50, 'Fragrant purple flower, ideal for borders and pots.', 'Plant', 7.99),
6 ('Watering Can', 30, '5-litre metal watering can with long spout.', 'Tool', 12.50),
7 ('Compost Bin', 15, 'Eco-friendly compost bin made from recycled plastic.', 'Accessory', 45.00),
8 ('Tomato Seed Pack', 100, 'Contains 50 high-yield tomato seeds.', 'Plant', 2.99),
9 ('Hand Trowel', 40, 'Stainless steel blade with wooden handle.', 'Tool', 6.50),
10 ('Organic Fertilizer', 25, 'Slow-release pellets for flower beds and lawns.', 'Accessory', 9.75);
  
```

The screenshot shows a SQL editor with the above code. Below the editor is a 'Result Grid' displaying the inserted data:

ProductID	ProductName	ProductQuantity	ProductDescription	ProductType	ProductPrice
1	Lavender Plant	50	Fragrant purple flower, ideal for borders and pots.	Plant	7.99
2	Watering Can	30	5-litre metal watering can with long spout.	Tool	12.50
3	Compost Bin	15	Eco-friendly compost bin made from recycled plastic.	Accessory	45.00
4	Tomato Seed Pack	100	Contains 50 high-yield tomato seeds.	Plant	2.99
5	Hand Trowel	40	Stainless steel blade with wooden handle.	Tool	6.50
6	Organic Fertilizer	25	Slow-release pellets for flower beds and lawns.	Accessory	9.75

Fig. 3.7 inserting into inventory table

5. Sales table

Creation:

```

1 • CREATE TABLE Sales (
2     SaleID INT AUTO_INCREMENT PRIMARY KEY,
3     CustomerID INT,
4     EmployeeID INT,
5     ItemPurchased VARCHAR(100),
6     SaleQuantity INT,
7     SalePrice DECIMAL(10, 2),
8     SaleDate DATE,
9     FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
10    FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)
11 );
12 • describe table sales
  
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	sales		ALL					1	100.00	

Fig. 3.8 creating sales table

Insertion:

```

1 • SELECT * FROM bloom_field_garden_centre.sales;
2 • INSERT INTO Sales (
3     SaleID, CustomerID, EmployeeID, ItemPurchased, SaleQuantity, SalePrice, SaleDate
4 ) VALUES
5     (1, 3, 2, 'Lavender Plant', 10, 15.98, '2025-03-12'),
6     (2, 2, 3, 'Compost Bin', 14, 45.00, '2025-04-05'),
7     (3, 3, 1, 'Daffodil Bunch', 13, 15.00, '2025-03-30'),
8     (4, 5, 3, 'Raised Bed Kit', 4, 30.00, '2025-04-10'),
9     (5, 2, 5, 'Basil Plant', 10, 5.25, '2025-04-15');
10
  
```

SaleID	CustomerID	EmployeeID	ItemPurchased	SaleQuantity	SalePrice	SaleDate
1	3	2	Lavender Plant	10	15.98	2025-03-12
2	2	3	Compost Bin	14	45.00	2025-04-05
3	3	1	Daffodil Bunch	13	15.00	2025-03-30
4	5	3	Raised Bed Kit	4	30.00	2025-04-10
5	2	5	Basil Plant	10	5.25	2025-04-15

Fig.3.9 inserting into sales

6. Training and records

Creation:

```

1 • CREATE TABLE TrainingAndRecords (
2     TraineeID INT AUTO_INCREMENT PRIMARY KEY,
3     EmployeeID INT,
4     TrainingDuration VARCHAR(50),
5     AccomplishmentDate DATE,
6     TrainingCertification VARCHAR(100),
7     TrainingDescription TEXT,
8     FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)
9 );
10 • describe table TrainingAndRecords
11

```

Result Grid

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	TrainingAndRecords		ALL					1	100.00	

Fig. 3.10 creating training and records table

Insertion:

```

1 • SELECT * FROM bloom_field_garden_centre.trainingandrecords;
2 • INSERT INTO TrainingAndRecords (
3     EmployeeID, TrainingDuration, AccomplishmentDate,
4     TrainingCertification, TrainingDescription
5 ) VALUES
6 (1, '3 days', '2025-01-15', 'Certified Plant Specialist', 'Advanced plant care and identification techniques.'),
7 (2, '5 days', '2024-11-10', 'Inventory Management Level 2', 'Training on stock rotation, storage safety, and digital inventory systems.'),
8 (3, '2 days', '2025-02-20', 'Workshop Facilitation Certificate', 'Interactive sessions on leading garden workshops.'),
9 (4, '1 day', '2024-12-01', 'Customer Service Essentials', 'Customer interaction, conflict resolution, and communication skills.'),
10 (5, '4 days', '2025-03-05', 'Tool Maintenance and Safety', 'Safe use and upkeep of gardening tools and machinery.'),
11

```

Result Grid

TraineeID	EmployeeID	TrainingDuration	AccomplishmentDate	TrainingCertification	TrainingDescription
1	1	3 days	2025-01-15	Certified Plant Specialist	Advanced plant care and identification techniques.
2	2	5 days	2024-11-10	Inventory Management Level 2	Training on stock rotation, storage safety, and digital inventory systems.
3	3	2 days	2025-02-20	Workshop Facilitation Certificate	Interactive sessions on leading garden workshops.
4	4	1 day	2024-12-01	Customer Service Essentials	Customer interaction, conflict resolution, and communication skills.
5	5	4 days	2025-03-05	Tool Maintenance and Safety	Safe use and upkeep of gardening tools and machinery.

Fig.3.11 inserting data into training records

7. Events

Creation:

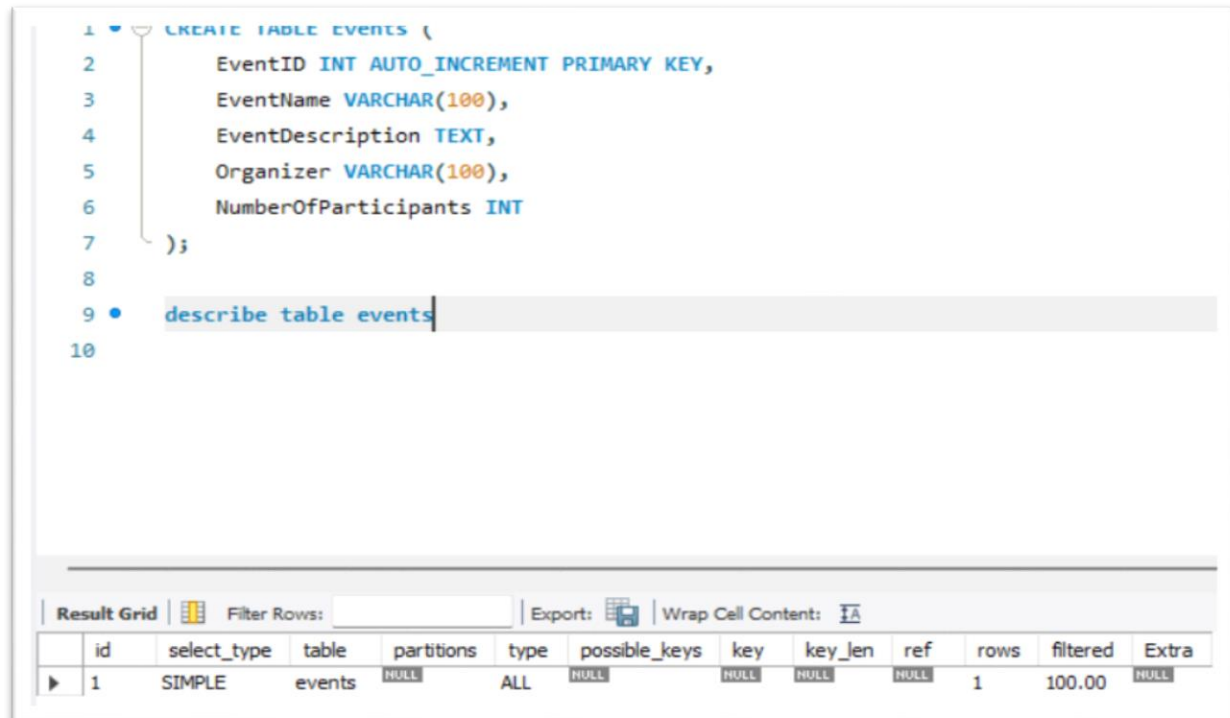


Fig. 3.12 creating table of event

Insertion:

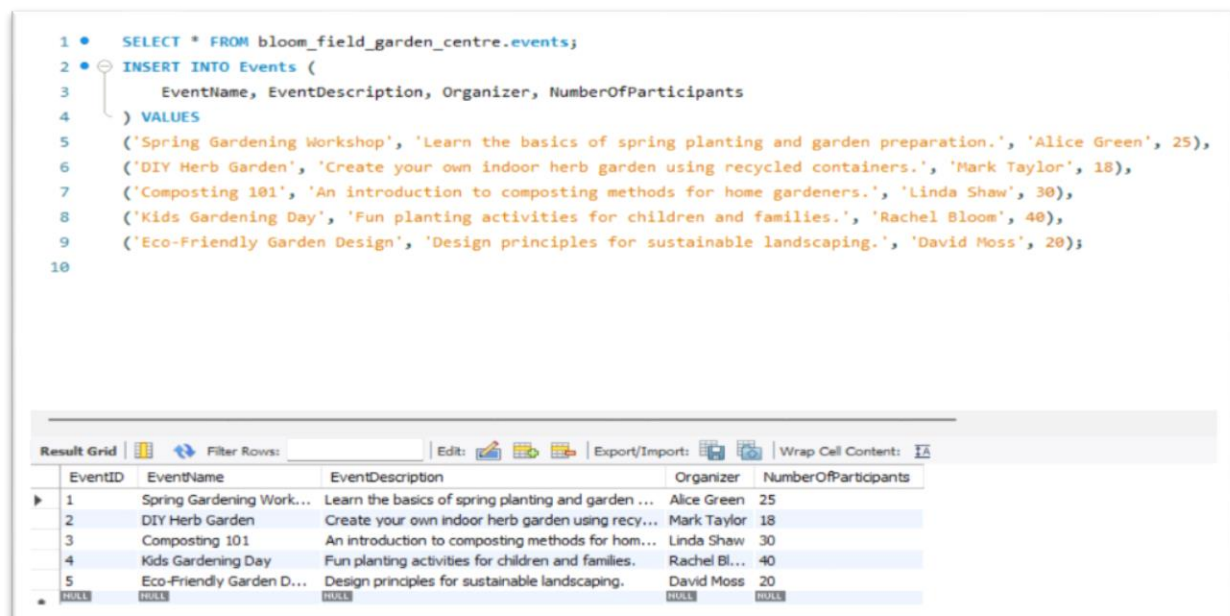
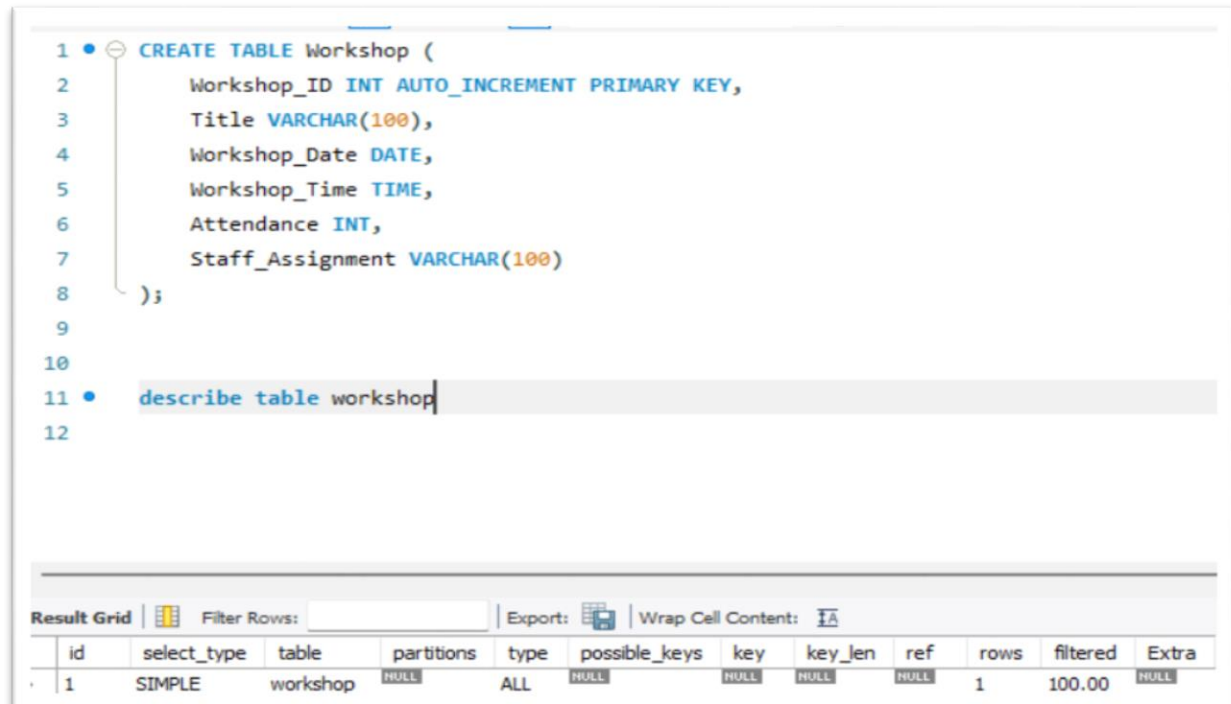


Fig. 3.13 inserting data into events

8. Workshop

Creation:



```

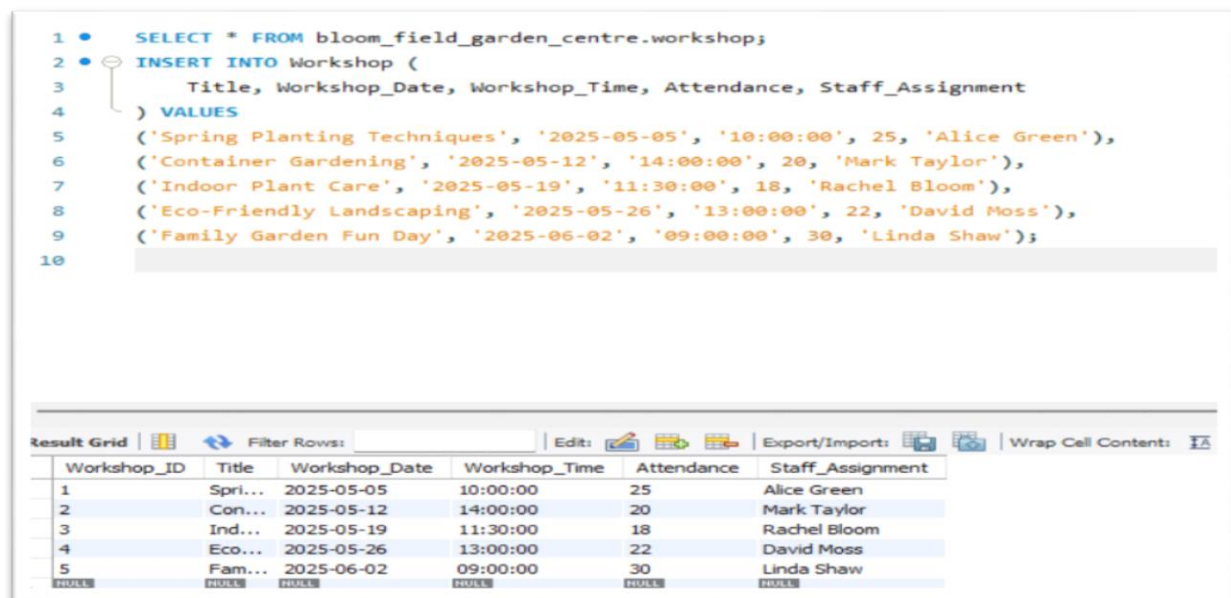
1 • CREATE TABLE Workshop (
2     Workshop_ID INT AUTO_INCREMENT PRIMARY KEY,
3     Title VARCHAR(100),
4     Workshop_Date DATE,
5     Workshop_Time TIME,
6     Attendance INT,
7     Staff_Assignment VARCHAR(100)
8 );
9
10
11 • describe table workshop
12

```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	workshop		ALL					1	100.00	

Fig. 3.14 creating table of workshop

Insertion:



```

1 • SELECT * FROM bloom_field_garden_centre.workshop;
2 • INSERT INTO Workshop (
3     Title, Workshop_Date, Workshop_Time, Attendance, Staff_Assignment
4 ) VALUES
5     ('Spring Planting Techniques', '2025-05-05', '10:00:00', 25, 'Alice Green'),
6     ('Container Gardening', '2025-05-12', '14:00:00', 20, 'Mark Taylor'),
7     ('Indoor Plant Care', '2025-05-19', '11:30:00', 18, 'Rachel Bloom'),
8     ('Eco-Friendly Landscaping', '2025-05-26', '13:00:00', 22, 'David Moss'),
9     ('Family Garden Fun Day', '2025-06-02', '09:00:00', 30, 'Linda Shaw');
10

```

Workshop_ID	Title	Workshop_Date	Workshop_Time	Attendance	Staff_Assignment
1	Spr...	2025-05-05	10:00:00	25	Alice Green
2	Con...	2025-05-12	14:00:00	20	Mark Taylor
3	Ind...	2025-05-19	11:30:00	18	Rachel Bloom
4	Eco...	2025-05-26	13:00:00	22	David Moss
5	Fam...	2025-06-02	09:00:00	30	Linda Shaw

Fig. 3.15 inserting data into workshop

Task 4 – 15 Marks Role-based Access Control

a) Implement role-based access mechanisms:

- Creation of roles:

```
1 CREATE ROLE sales_role;  
2 CREATE ROLE admin_role;
```

Fig 4.1: Role Creation for (RBAC)

- Creation of users for each role:

```
1 CREATE USER 'sales_role_user'@'localhost' IDENTIFIED BY 'Sales@123';  
2 CREATE USER 'admin_role_user'@'localhost' IDENTIFIED BY 'Admin@123';
```

Fig 4.2: User Creation for each role for (RBAC)

- Identification, Justification and Privilege Permissions for Assigned Tables for Each Role:

1. Role-1 (sales):

Chosen Tables:

- Inventory
- Sales Transactions

Assigned Privileges:

- SELECT** – Grants access to view product listings and transaction history (applies to both tables).
- INSERT** – Allows new entries to be added, such as registering a new sale or adding new product arrivals.
- UPDATE** – Permitted only on the Inventory table, for updating stock quantities when inventory is restocked.

Justification:

Sales team members are directly involved in managing product sales and stock tracking. They must be able to view available items and past transactions, record new purchases made by customers, and update the product stock accordingly. Update access is restricted only to the inventory table to prevent alteration of completed transactions. This setup supports efficient day-to-day sales operations while maintaining data accuracy and integrity.

2. Role-2 (admin):

Chosen Tables:

- Employee
- Training_and_Records

Assigned Privileges:

1. **All Privileges** – Including SELECT, INSERT, UPDATE, and DELETE.

Justification:

Administrators are tasked with managing internal staff information and their training records. These tables require full access so that HR or admin users can create new employee profiles, assign or update roles, log training progress, and remove outdated or incorrect records when necessary. Granting full control to these users ensures smooth administrative functioning, consistent data management, and accountability over human resources.

- **Granting User-Permissions:**

```
1 • GRANT sales TO 'sales_role_user'@'localhost';
2   GRANT admin TO 'admin_role_user'@'localhost';
```

Figure 4.3 Granting User-Permissions

- **Granting permissions to both the roles:**

1. Role 1 (Sales):

```
1 • GRANT SELECT, INSERT ON Sales TO sales;
2   GRANT SELECT ON Employees TO sales;
```

Fig 4.4: Granting Permissions to the sales role

2. Role 2 (Admin):

```
1 • GRANT ALL PRIVILEGES ON Sales TO admin;
2   GRANT ALL PRIVILEGES ON Employees TO admin;
```

Fig 4.5: Granting Permissions to the admin role

- **Output:**

1. Verification of access permissions for access role:

```
1 • show grants for 'sales_role_user'@'localhost';
```

Fig 4.6: Query for displaying sales role access permissions

Grants for sales_role_user@localhost	
▶	GRANT USAGE ON *.* TO 'sales_role_user'@'localhost'
	GRANT 'sales'@'%' TO 'sales_role_user'@'localhost'

Fig 4.7: Displaying output of sales role access permissions

2. Tables that the sales role is permitted to access and the specific permissions granted:

```

1  SELECT
2      'INVENTORY' AS TABLE_Name,
3      'SELECT, UPDATE' AS Allowed_Operations
4  UNION
5  SELECT
6      'CUSTOMERS' AS TABLE_Name,
7      'SELECT' AS Allowed_Operations;

```

Fig 4.8: Query to display allowed tables and their access permission for sales role

	TABLE_Name	Allowed_Operations
▶	INVENTORY	SELECT, UPDATE
	CUSTOMERS	SELECT

Fig 4.9: Displaying tables allowed to the sales role

3. Tables that the admin role is permitted to access and the specific permissions granted

```

1  •  SELECT
2      'INVENTORY' AS TABLE_Name,
3      'ALL PRIVILEGES' AS Allowed_Operations
4  UNION
5  SELECT
6      'EMPLOYEE' AS TABLE_Name,
7      'ALL PRIVILEGES' AS Allowed_Operations;

```

Fig 4.10: Query to display allowed tables and their access permission for sales role

	TABLE_Name	Allowed_Operations
▶	INVENTORY	ALL PRIVILEGES
	EMPLOYEE	ALL PRIVILEGES

Fig 4.11: Displaying tables allowed to the admin role

Task 5 – 20 Marks SQL queries

- a) Write a SQL query to update the inventory to reflect the latest pricing after a supplier price change. This query increases the price of all gardening tools by 10% to adjust for an increase in supplier costs.

1. Query

```
1  UPDATE Inventory
2  SET ProductPrice = ProductPrice * 1.10
3  WHERE ProductType = 'Tool';
4  |
```

Fig 5.1: Query for updating Inventory

2. outcome

	ProductID	ProductName	ProductQuantity	ProductDescription	ProductType	ProductPrice
▶	1	Lavender Plant	50	Fragrant purple flower, ideal for borders and pots.	Plant	7.99
	2	Watering Can	30	5-litre metal watering can with long spout.	Tool	15.13
	3	Compost Bin	15	Eco-friendly compost bin made from recycled plastic.	Accessory	45.00
	4	Tomato Seed Pack	100	Contains 50 high-yield tomato seeds.	Plant	2.99
	5	Hand Trowel	40	Stainless steel blade with wooden handle.	Tool	7.87
	6	Organic Fertilizer	25	Slow-release pellets for flower beds and lawns.	Accessory	9.75
+	NULL	NULL	NULL	NULL	NULL	NULL

Fig 5.2: Output for updated inventory

b) Write a SQL query to retrieve the highest priced item from each of the following categories: plants, gardening tools and accessories. Provide the product name, type, price and a brief description for each.

1. Query

```
1  SELECT ProductName, ProductType, ProductPrice, ProductDescription
2  FROM Inventory i
3  WHERE ProductPrice = (
4      SELECT MAX(ProductPrice)
5      FROM Inventory
6      WHERE ProductType = i.ProductType
7  )
8  AND ProductType IN ('Plant', 'Tool', 'Accessory');
```

Fig 5.3: Query for retrieving highest price from every category

2. Outcome:

	ProductName	ProductType	ProductPrice	ProductDescription
►	Lavender Plant	Plant	7.99	Fragrant purple flower, ideal for borders and pots.
	Watering Can	Tool	13.75	5-litre metal watering can with long spout.
	Compost Bin	Accessory	45.00	Eco-friendly compost bin made from recycled plastic.

Fig 5.4: Output for retrieving data of highest priced from every category

c) Write a SQL query to summarise the total sales transactions for a specific past month, breaking down the data by item category, including quantities sold and total revenue per category.

1. Query

```

1 • SELECT
2     i.ProductType,
3     SUM(s.SaleQuantity) AS TotalQuantitySold,
4     SUM(s.SalePrice) AS TotalRevenue
5 FROM Sales s
6 JOIN Inventory i ON s.ItemPurchased = i.ProductName
7 WHERE MONTH(SaleDate) = 3 AND YEAR(SaleDate) = 2025 -- Adjust month/year as needed
8 GROUP BY i.ProductType;

```

Fig 5.5: Query for summarizing the total sales transactions for a specific period

2. Outcome

	ProductType	TotalQuantitySold	TotalRevenue
▶	Plant	5	24.95

Fig 5.6: Output for summarizing the total sales transactions of a specific period

d) Write a SQL query to find all upcoming workshops and events within the next month, including the titles, dates, times, and the current number of registered participants

1. Query:

```

1 • SELECT Title, Workshop_Date, Workshop_Time, Attendance
2 FROM Workshop
3 WHERE Workshop_Date BETWEEN CURDATE() AND DATE_ADD(CURDATE(), INTERVAL 1 MONTH);

```

Fig 5.7: Query to find all upcoming events and workshops

2. Outcome:

	Title	Workshop_Date	Workshop_Time	Attendance
▶	Spring Planting Techniques	2025-05-05	10:00:00	25
	Container Gardening	2025-05-12	14:00:00	20

Fig 5.8: Output for all upcoming workshops and events

e) Write a SQL query to retrieve and rank employees based on the total revenue they've generated from these transactions up until a specific date.

1. Query

```
1 SELECT
2     e.EmployeeName,
3     SUM(s.SalePrice) AS TotalRevenue
4 FROM Sales s
5 JOIN Employees e ON s.EmployeeID = e.EmployeeID
6 WHERE s.SaleDate <= '2025-04-15' -- Replace with desired cutoff date
7 GROUP BY e.EmployeeName
8 ORDER BY TotalRevenue DESC;
```

Fig 5.9: Query for ranking employees based on the total revenue

2. Outcome:

	EmployeeName	TotalRevenue
►	Linda Chen	75.00
	Mark Evans	15.98
	Sarah Johnson	15.00
	Emily Moore	5.25

Fig 5.10: Output that ranks employees based on their total revenue

References:

1. Oracle. (2024). *MySQL 8.0 Reference Manual*. [online] Available at: <https://dev.mysql.com/doc/refman/8.0/en/> [Accessed 16 Apr. 2025].
2. W3Schools. (n.d.). *SQL Tutorial*. [online] Available at: <https://www.w3schools.com/sql/> [Accessed 16 Apr. 2025].
3. Stack Overflow. (n.d.). *Common MySQL Error Codes (e.g. Error 1396)*. [online forum] Available at: <https://stackoverflow.com/> [Accessed 16 Apr. 2025].
4. GeeksforGeeks. (n.d.). *SQL Queries Examples*. [online] Available at: <https://www.geeksforgeeks.org/sql-queries-examples/> [Accessed 16 Apr. 2025].
5. TutorialsPoint. (n.d.). *SQL – GRANT and REVOKE*. [online] Available at: <https://www.tutorialspoint.com/sql/sql-grant-revoke.htm> [Accessed 16 Apr. 2025].
6. IBM. (n.d.). *Database Roles and Permissions*. [online] Available at: <https://www.ibm.com/docs/en/db2> [Accessed 16 Apr. 2025].

Candidate checklist

Please use the following checklist to ensure that your work is ready for submission.

Have you read the NCC Education document *Academic Misconduct Policy* and ensured that you have acknowledged all the sources that you have used in your work? ☐

Have you completed the *Statement and Confirmation of Own Work* form and attached it to your assignment? **You must do this.** ☐

Have you ensured that your work does not contain viruses and can be run directly? ☐