

Experiment 2: DDL Commands

AIM

To study and implement DDL commands and different types of constraints.

THEORY

1. CREATE

Used to create a new relation (table).

Syntax:

```
CREATE TABLE (  
    field_1 data_type(size),  
    field_2 data_type(size),  
    ...  
);
```



2. ALTER

Used to add, modify, drop, or rename fields in an existing relation. (a) ADD

```
ALTER TABLE std ADD (Address CHAR(10));
```



(b) MODIFY

```
ALTER TABLE relation_name MODIFY (field_1 new_data_type(size));
```



(c) DROP

```
ALTER TABLE relation_name DROP COLUMN field_name;
```



(d) RENAME

```
ALTER TABLE relation_name RENAME COLUMN old_field_name TO new_field_name;
```



3. DROP TABLE

Used to permanently delete the structure and data of a table.

```
DROP TABLE relation_name;
```



4. RENAME

Used to rename an existing database object.

```
RENAME TABLE old_relation_name TO new_relation_name;
```



CONSTRAINTS

Constraints are used to specify rules for the data in a table. If there is any violation between the constraint and the data action, the action is aborted by the constraint. It can be specified when the table is created (using CREATE TABLE) or after it is created (using ALTER TABLE).

1. NOT NULL

When a column is defined as NOT NULL, it becomes mandatory to enter a value in that column. Syntax:

```
CREATE TABLE Table_Name (  
    column_name data_type(size) NOT NULL  
);
```



2. UNIQUE

Ensures that values in a column are unique. Syntax:

```
CREATE TABLE Table_Name (  
    column_name data_type(size) UNIQUE  
);
```



3. CHECK

Specifies a condition that each row must satisfy. Syntax:

```
CREATE TABLE Table_Name (  
    column_name data_type(size) CHECK (logical_expression)  
);
```



4. PRIMARY KEY

Used to uniquely identify each record in a table. Properties: Must contain unique values. Cannot be null. Should contain minimal fields. Syntax:

```
CREATE TABLE Table_Name (  
    column_name data_type(size) PRIMARY KEY  
);
```



5. FOREIGN KEY

Used to reference the primary key of another table. Syntax:

```
CREATE TABLE Table_Name (  
    column_name data_type(size),  
    FOREIGN KEY (column_name) REFERENCES other_table(column)  
);
```



6. DEFAULT

Used to insert a default value into a column if no value is specified.

Syntax:

```
CREATE TABLE Table_Name (  
    col_name1 data_type,  
    col_name2 data_type,  
    col_name3 data_type DEFAULT 'default_value'  
);
```



Question 1

Create a table named `Products` with the following constraints:

- `ProductID` as `INTEGER` should be the primary key.
- `ProductName` as `TEXT` should be unique and not `NULL`.
- `Price` as `REAL` should be greater than 0.
- `StockQuantity` as `INTEGER` should be non-negative.

For example:

Test	Result			
INSERT INTO Products (ProductID, ProductName, Price, StockQuantity) VALUES (1, 'Laptop', 999.99, 10); select * from Products;	ProductID	ProductName	Price	StockQuantity
	-----	-----	-----	-----
	1	Laptop	999.99	10

```
--  
INSERT INTO Employee(EmployeeID,Name,Position)  
values(5,          'George Clark',  'Consultant');  
  
INSERT INTO Employee(EmployeeID,Name,Position,Department,Salary)  
values(7,          'Noah Davis',    'Manager',    'HR',          60000);  
  
INSERT INTO Employee(EmployeeID,Name,Position,Department)  
values(8,          'Ava Miller',    'Consultant', 'IT');
```



Output:

Test	Expected	Got																									
✓ INSERT INTO Products (ProductID, ProductName, Price, StockQuantity) VALUES (1, 'Laptop', 999.99, 10); select * from Products;	<table><tr><th>ProductID</th><th>ProductName</th><th>Price</th><th>StockQuantity</th></tr><tr><td>-----</td><td>-----</td><td>-----</td><td>-----</td></tr><tr><td>1</td><td>Laptop</td><td>999.99</td><td>10</td></tr></table>	ProductID	ProductName	Price	StockQuantity	-----	-----	-----	-----	1	Laptop	999.99	10	<table><tr><th>ProductID</th><th>ProductName</th><th>Price</th><th>StockQuantity</th></tr><tr><td>-----</td><td>-----</td><td>-----</td><td>-----</td></tr><tr><td>1</td><td>Laptop</td><td>999.99</td><td>10</td></tr></table>	ProductID	ProductName	Price	StockQuantity	-----	-----	-----	-----	1	Laptop	999.99	10	✓
ProductID	ProductName	Price	StockQuantity																								
-----	-----	-----	-----																								
1	Laptop	999.99	10																								
ProductID	ProductName	Price	StockQuantity																								
-----	-----	-----	-----																								
1	Laptop	999.99	10																								
✓ INSERT INTO Products (ProductID, ProductName, Price, StockQuantity) VALUES (2, 'Smartphone', 200, 50),(3, 'Smartphone', 100, 50);	Error: UNIQUE constraint failed: Products.ProductName	Error: UNIQUE constraint failed: Products.ProductName	✓																								

Passed all tests! ✓

Correct
Marks for this submission: 10.00/10.00.

Question 2

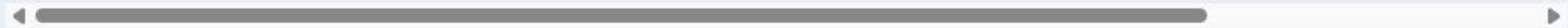
--

Create a table named `Attendance` with the following constraints:

- `AttendanceID` as `INTEGER` should be the primary key.
- `EmployeeID` as `INTEGER` should be a foreign key referencing `Employees(EmployeeID)`.
- `AttendanceDate` as `DATE`.
- `Status` as `TEXT` should be one of 'Present', 'Absent', 'Leave'.

For example:

Test	Result						
INSERT INTO Attendance (AttendanceID, EmployeeID, AttendanceDate, Status) VALUES (1, 1, '2024-08-01', 'Present'); SELECT * FROM Attendance;	<table><tr><td>AttendanceID</td><td>Em</td></tr><tr><td>-----</td><td>--</td></tr><tr><td>1</td><td>1</td></tr></table>	AttendanceID	Em	-----	--	1	1
AttendanceID	Em						
-----	--						
1	1						



```
--
CREATE TABLE Attendance (
  AttendanceID INTEGER PRIMARY KEY,
  EmployeeID INTEGER,
  AttendanceDate DATE,
  Status TEXT CHECK (Status IN ('Present', 'Absent', 'Leave')),
  FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)
);
```

Output:



	Expected	Got	
01', 'Present'); ----- 1 1 2024-08-01 Present	AttendanceID EmployeeID AttendanceDate Status ----- 1 1 2024-08-01 Present	AttendanceID EmployeeID AttendanceDate Status ----- 1 1 2024-08-01 Present	✓
-03', 'Absent'); ----- Error: FOREIGN KEY constraint failed	Error: FOREIGN KEY constraint failed	Error: FOREIGN KEY constraint failed	✓

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 3

--

Write a SQL Query to add attribute Date_of_joining as Date and rename the attribute job_title as Designation in the table 'Employees'

For example:

Test	Result					
pragma table_info('Employees');	cid	name	type	notnull	dflt_value	pk
	-----	-----	-----	-----	-----	-----
	0	employee_id	INT	0		1
	1	first_name	VARCHAR(50	0		0
	2	last_name	VARCHAR(50	0		0
	3	Designation	VARCHAR(10	0		0
	4	Date_of_joi	Date	0		0



--

```
ALTER TABLE Employees
ADD COLUMN Date_of_joining Date;
```

```
ALTER TABLE Employees
RENAME COLUMN job_title To Designation;
```

Output:

	Expected						Got				
) ;	cid	name	type	notnull	dflt_value	pk	cid	name	type	notnull	dflt_
	0	employee_id	INT	0		1	0	employee_id	INT	0	
	1	first_name	VARCHAR(50	0		0	1	first_name	VARCHAR(50	0	
	2	last_name	VARCHAR(50	0		0	2	last_name	VARCHAR(50	0	
	3	Designation	VARCHAR(10	0		0	3	Designation	VARCHAR(10	0	
	4	Date_of_joi	Date	0		0	4	Date_of_joi	Date	0	

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 4

Question **4**

Correct

Mark 10.00 out of 10.00

🚩 Flag question

create a table named jobs including columns job_id, job_title, min_salary and max_salary, and make sure that, the default value for job_title is blank and min_salary is 8000 and max_salary is NULL will be entered automatically at the time of insertion if no value assigned for the specified columns.

For example:

Test	Result	
INSERT INTO jobs (job_id, job_title, min_salary, max_salary) VALUES (1, 'Software Engineer', 9000, 15000); SELECT * FROM jobs;	job_id	job_title
	-----	-----
	1	Software En

Answer: (penalty regime: 0 %)

```
CREATE TABLE jobs (  
  job_id INTEGER PRIMARY KEY,  
  job_title TEXT NOT NULL DEFAULT '',  
  min_salary INTEGER NOT NULL DEFAULT 8000,  
  max_salary INTEGER DEFAULT NULL  
);
```



Output:

	Expected				Got				
9000, 15000));	job_id	job_title	min_salary	max_salary	job_id	job_title	min_salary	max_salary	✓
	-----	-----	-----	-----	-----	-----	-----	-----	
	1	Software Engineer	9000	15000	1	Software Engineer	9000	15000	
	job_id	job_title	min_salary	max_salary	job_id	job_title	min_salary	max_salary	✓
	-----	-----	-----	-----	-----	-----	-----	-----	
	2	Data Scientist	8000		2	Data Scientist	8000		



Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 5

--

Question **5**

Correct

Mark 10.00 out of 10.00

Flag question

Create a table named **Departments** with the following columns:

- **DepartmentID** as **INTEGER**
- **DepartmentName** as **TEXT**

For example:

Test	Result					
pragma table_info('Departments');	cid	name	type	notnull	dflt_value	pk
	-----	-----	-----	-----	-----	-----
	0	DepartmentID	INTEGER	0		0
	1	DepartmentName	TEXT	0		0



```
--  
CREATE TABLE Departments(  
  DepartmentID INTEGER,  
  DepartmentName TEXT  
);
```

Output:

Expected						Got				
cid	name	type	notnull	dflt_value	pk	cid	name	type	notnull	dflt_value
0	DepartmentID	INTEGER	0		0	0	DepartmentID	INTEGER	0	
1	DepartmentName	TEXT	0		0	1	DepartmentName	TEXT	0	
Depar DepartmentName						Depar DepartmentName				
-----						-----				
2						2				

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 6

--

Insert all books from `Out_of_print_books` into `Books`
Table attributes are `ISBN`, `Title`, `Author`, `Publisher`, `YearPublished`

For example:

Test	Result				
select * from Books;	ISBN	Title	Author	Publisher	YearPublished
	-----	-----	-----	-----	-----
	978-1234567890	The Lost World	Arthur Conan Doyle	Vintage Books	1912
	978-0987654321	Gone with the	Margaret Mitchell	Macmillan	1936
	978-1122334455	Moby Dick	Herman Melville	Harper & Brot	1851

--

```
select *from Out_of_print_books
union all
select *from Books
```



Output:

Expected					Got		
ISBN	Title	Author	Publisher	YearPublished	ISBN	Title	Author
978-1234567890	The Lost World	Arthur Conan Doyle	Vintage Books	1912	978-1234567890	The Lost World	Arthur Conan D
978-0987654321	Gone with the	Margaret Mitchell	Macmillan	1936	978-0987654321	Gone with the	Margaret Mitch
978-1122334455	Moby Dick	Herman Melville	Harper & Brot	1851	978-1122334455	Moby Dick	Herman Melvill

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 7

--

Create a new table named `item` with the following specifications and constraints:

1. `item_id` as TEXT and as primary key.
2. `item_desc` as TEXT.
3. `rate` as INTEGER.
4. `icom_id` as TEXT with a length of 4.
5. `icom_id` is a foreign key referencing `com_id` in the `company` table.
6. The foreign key should cascade updates and deletes.
7. `item_desc` and `rate` should not accept NULL.

For example:

Test	Result			
INSERT INTO item VALUES("ITM5","Charlie Gold",700,"COM4"); UPDATE company SET com_id='COM5' WHERE com_id='COM4'; SELECT * FROM item;	item_id	item_desc	rate	icom_id
	-----	-----	-----	-----
	ITM5	Charlie Gold	700	COM5

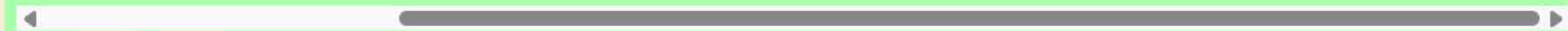
--

```
CREATE TABLE item (  
    item_id TEXT PRIMARY KEY,  
    item_desc TEXT NOT NULL,  
    rate INTEGER NOT NULL,  
    icom_id TEXT CHECK(4),  
    FOREIGN KEY (icom_id) REFERENCES company(com_id)  
    ON UPDATE CASCADE  
    ON DELETE CASCADE  
);
```



Output:

	Expected				Got				
arlie Gold",700,"COM4"); HERE com_id='COM4';	item_id	item_desc	rate	icom_id	item_id	item_desc	rate	icom_id	✓
	-----	-----	-----	-----	-----	-----	-----	-----	
	ITM5	Charlie Gold	700	COM5	ITM5	Charlie Gold	700	COM5	
a-n-Nut",14,"COM2"); arlie Gold",700,"COM4"); COM2';	item_id	item_desc	rate	icom_id	item_id	item_desc	rate	icom_id	✓
	-----	-----	-----	-----	-----	-----	-----	-----	
	ITM5	Charlie Gold	700	COM4	ITM5	Charlie Gold	700	COM4	



Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 8

--

Write a SQL Query to add an attribute **designation** in the **employee** table with the data type **VARCHAR(50)**.

For example:

Test	Result					
pragma table_info('employee');	cid	name	type	notnull	dflt_value	pk
	-----	-----	-----	-----	-----	-----
	0	id	integer	0		0
	1	salary	number	0		0
	2	designatio	varchar(50	0		0



--

```
ALTER TABLE employee  
ADD COLUMN designation varchar(50);
```

Output:

Expected						Got				
cid	name	type	notnull	dflt_value	pk	cid	name	type	notnull	dflt_value
0	id	integer	0		0	0	id	integer	0	
1	salary	number	0		0	1	salary	number	0	
2	designatio	varchar(50	0		0	2	designatio	varchar(50	0	

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 9

--

Insert a product with ProductID 104, Name Tablet, and Category Electronics into the Products table, where Price and Stock should use default values.

For example:

Test	Result				
SELECT ProductID, Name, Category, Price, Stock FROM Products WHERE ProductID = 104;	ProductID	Name	Category	Price	Stock
	-----	-----	-----	-----	-----
	104	Tablet	Electronics	100	50

--

```
INSERT INTO Products (ProductID, Name, Category)
VALUES (104, 'Tablet', 'Electronics');
```



Output:

	Test	Expected					Got	
✓	SELECT ProductID, Name, Category, Price, Stock FROM Products WHERE ProductID = 104;	ProductID	Name	Category	Price	Stock	ProductID	Nam
		-----	-----	-----	-----	-----	-----	---
		104	Tablet	Electronics	100	50	104	Tab

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

Question 10

--

Create a table named **Employees** with the following constraints:

- **EmployeeID** should be the primary key.
- **FirstName** and **LastName** should be **NOT NULL**.
- **Email** should be unique.
- **Salary** should be greater than 0.
- **DepartmentID** should be a foreign key referencing the **Departments** table.

For example:

Test	Result
-- Attempt to insert a record with NULL FirstName INSERT INTO Employees (EmployeeID, FirstName, LastName, Email, Salary, DepartmentID) VALUES (1, NULL, 'Doe', 'john.doe@example.com', 50000, 1);	Error: NOT NULL constraint failed: Employees.

--

```
CREATE TABLE Employees(  
EmployeeID INTEGER primary key,  
FirstName INTEGER NOT NULL,  
LastName INTEGER NOT NULL,  
Email VARCHAR(50) unique,  
Salary CHECK (Salary>0),  
DepartmentID INTEGER,  
foreign key(DepartmentID) references Departments(DepartmentID)  
);
```



Output:

	Test	Expected
✓	-- Attempt to insert a record with NULL FirstName INSERT INTO Employees (EmployeeID, FirstName, LastName, Email, Salary, DepartmentID) VALUES (1, NULL, 'Doe', 'john.doe@example.com', 50000, 1);	Error: NOT NULL constraint failed: Emp
✓	-- Insert a valid record first INSERT INTO Employees (EmployeeID, FirstName, LastName, Email, Salary, DepartmentID) VALUES (2, 'Jane', 'Doe', 'jane.doe@example.com', 60000, 1); -- Attempt to insert a record with a duplicate Email INSERT INTO Employees (EmployeeID, FirstName, LastName, Email, Salary, DepartmentID) VALUES (3, 'Janet', 'Smith', 'jane.doe@example.com', 70000, 2);	Error: UNIQUE constraint failed: Emplo

Passed all tests! ✓

Correct

Marks for this submission: 10.00/10.00.

RESULT

Thus, the SQL queries to implement different types of constraints and DDL commands have been executed successfully.