SQL QnA (DDL Constraints):-

BASIC:-

Q1:-

```
CREATE TABLE students (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL UNIQUE
);

INSERT INTO students () VALUES ();
INSERT INTO students () VALUES ();
```

ANS:-

First entry = pass, as 1 in id and ‘’ in name.
Second entry = fail, as 2 in id but in name, as it was empty, ‘’ was to be added but it already is, and the constraint being UNIQUE, doesn’t allow duplicate (‘’) values.

Q2:-

```
CREATE TABLE employees (
    emp_id INT PRIMARY KEY,
    email VARCHAR(255) NOT NULL UNIQUE,
    dept_id INT,
    FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
);

INSERT INTO employees (emp_id, email, dept_id) VALUES (101, 'john@example.com', 1);
```

ANS:-

RESTRICT referential action takes place, as you can’t insert the value in FK before inserting the same value in PK.

Q3:-

```
CREATE TABLE products (
    id INT PRIMARY KEY,
    price DECIMAL(10,2) CHECK (price > 0),
    stock INT DEFAULT 0
);
```

INSERT INTO products (id, price) VALUES (1, -50.00);

ANS:-
Constraint check failed, as values must be greater than 0, here it is -50.

Q4:-

```
CREATE TABLE users (
   user_id INT NOT NULL,
   username VARCHAR(50) UNIQUE,
   PRIMARY KEY (user_id)
);

INSERT INTO users (user_id, username) VALUES (1, NULL);
```

ANS:-
NO errors. In the attribute jaha pe sirf UNIQUE constraint laga hua hai, it can accept as many NULL values as it wants.
Logic:-
NULL = unknown, can be any value, so assumed that it will be unique, hence no error thrown.

Q5:-

```
CREATE TABLE orders (
   order_id INT AUTO_INCREMENT PRIMARY KEY,
   customer_name VARCHAR(100) NOT NULL DEFAULT 'Guest'
);

INSERT INTO orders () VALUES ();
INSERT INTO orders () VALUES ('');
```

ANS:-
First insert = 1, ‘’
Second insert = 2, ‘’
This is because you manually added ‘’. Agar aise hi choda hota empty, then the value would have been the default value ‘Guest’.

Q6:-

```
CREATE TABLE inventory (
   item_id INT PRIMARY KEY,
   quantity INT CHECK (quantity BETWEEN 1 AND 100)
```

);

INSERT INTO inventory (item_id, quantity) VALUES (1, NULL);

ANS:-
Pass. after inserting, we have 1, NULL.
This is because when you have check, it only has 2 ways. Either condition TRUE and pass, or FALSE and fail to pass. NULL does not give either TRUE or FALSE. Hence, it passes the condition.

Q7:-
```
CREATE TABLE logins (
   id INT AUTO_INCREMENT,
   username VARCHAR(50) NOT NULL,
   PRIMARY KEY (id, username)
);

INSERT INTO logins (username) VALUES ('alice');
INSERT INTO logins (username) VALUES ('alice');
```

ANS:-
1, alice
2, alice
No UNIQUE constraint applied.

ADVANCED:-

Q1:-
```
CREATE TABLE enrollments (
  student_id INT,
  course_id INT,
  PRIMARY KEY (student_id, course_id)
);

INSERT INTO enrollments VALUES (NULL, 101);
```

ANS:-
FAIL. this is because it is a composite pk, which is a PK, and does not allow NULL or DUPLICATES.

Q2:-

```
CREATE TABLE employees (
  email VARCHAR(255) UNIQUE
);

INSERT INTO employees VALUES (NULL);
INSERT INTO employees VALUES (NULL);
```

ANS:-

As said earlier, if the constraint is only UNIQUE, we can add as many NULL values, so yes,
output:-
NULL
NULL

Q3:-

```
CREATE TABLE accounts (
  id INT PRIMARY KEY AUTO_INCREMENT,
  status VARCHAR(50) NOT NULL DEFAULT 'active'
);

INSERT INTO accounts (id) VALUES (1);
INSERT INTO accounts (id, status) VALUES (2, NULL);
```

ANS:-

Output:-
1, active
2, error
2nd insertion failed because we can inserting NULL in the NOT NULL constraint applied attribute.

Q4:-

```
CREATE TABLE payments (
  id INT PRIMARY KEY,
  amount DECIMAL(10,2) CHECK (amount > 0)
);

INSERT INTO payments VALUES (1, NULL);
```

ANS:-

Output:-
1,NULL
If null inserted and constraint is just check, null easily passes through.

Q5:-

```
CREATE TABLE departments (
  id INT PRIMARY KEY
);

CREATE TABLE employees (
  id INT PRIMARY KEY,
  dept_id INT,
  FOREIGN KEY (dept_id) REFERENCES departments(id)
);

INSERT INTO employees VALUES (1, NULL);
```

ANS:-
In the FK, you can never add non-null values in it before adding it to the PK. But, here we are adding NULL value to the FK, which is completely allowed.
Output:-
1, NULL

Q6:-

```
CREATE TABLE books (
  isbn VARCHAR(13) PRIMARY KEY,
  book_id INT AUTO_INCREMENT
);

INSERT INTO books (isbn) VALUES ('9781234567890');
```

ANS:-
Output:-
Isbn = '9781234567890'
Book_id = fail to auto increment.
NOTE:-
AUTO_INCREMENT only works when the constraint applied to the attribute is PK or UNIQUE.

Q7:-

```
CREATE TABLE users (
  id INT PRIMARY KEY,
```

```
  age INT CHECK (age >= 18 AND age <= 60)
);

INSERT INTO users VALUES (1, 61);
INSERT INTO users VALUES (2, NULL);
```

ANS:-
First insert failed as 61 not passing check condition.
Second insert passed.

Q8:-

```
CREATE TABLE contacts (
  id INT PRIMARY KEY,
  phone VARCHAR(10) UNIQUE
);

INSERT INTO contacts VALUES (1, NULL);
INSERT INTO contacts VALUES (2, NULL);
INSERT INTO contacts VALUES (3, '1234567890');
INSERT INTO contacts VALUES (4, '1234567890');
```

ANS:-
4th insert failed as we are inserting duplicate values.
NOTE:-
NULL is not considered duplicate as i said, that NULL = Unknown, we don’t know what value it can be so we can’t directly classify that it will be a duplicate value.

Q9:-

```
CREATE TABLE products (
  product_id INT PRIMARY KEY AUTO_INCREMENT,
  sku VARCHAR(20) NOT NULL UNIQUE
);

INSERT INTO products (sku) VALUES (NULL);
```

ANS:-
Error, as constraint is NOT NULL.

Q10:-

```
CREATE TABLE authors (
  id INT PRIMARY KEY
```

```
);

CREATE TABLE books (
  id INT PRIMARY KEY,
  author_id INT,
  FOREIGN KEY (author_id) REFERENCES authors(id) ON DELETE SET NULL
);

INSERT INTO authors VALUES (1);
INSERT INTO books VALUES (10, 1);
DELETE FROM authors WHERE id = 1;
```

ANS:-

Authors:-

1

Books:-

10, 1

Authors me se id=1 nikala. So, in the referenced FK, the value becomes NULL.
