Rule 1: Information rule

In the current database, we see that all information based on the dental clinic relational database (including metadata) is stored in the form of rows and columns so that the anyone can easily perform operations on it with faster access i.e. arranging appointments.

PatNumber	PatientFirstName	Address
P01	Carine	345 View Ridge
P02	Susan	9408 Furth Circle
P03	Saoirse	14 Main Street

Sample table patient is given above. Here, the record of every patient number first name and address is stored in the table only. This will help Helen find patient records with ease.

MySQL

```
SELECT PatNumber, PatFirstName, Address FROM patient;
```

--Lists information on Patient Number, Patient First Name and address from those patients in the patient table.

Rule 2: Guaranteed Access

If Helen wishes to access unique patient records then the patient table must have a primary key which will be accessed through tableName and columnName. Suppose in mysql we want to fetch unique patient records then we will write queries like select PatNumber from patient. Here you will notice that PatNumber is a primary key which will fetch unique patients from the database. Use of this rule is very common in today's world database.

MySQL

```
SELECT PatNumber, PatFirstName, EirCode, City FROM patient WHERE City = 'Mallow';
```

-- Lists patient number, first name and Eircode in the patient table for those patients who are from Mallow.

Rule 3: Systematic treatment of NULL

Null has several meanings; it can mean missing data, not applicable or no value. It should be handled consistently. Some phone numbers and addresses from patient records are missing therefore they are assigned null values.

MySQL

```
SELECT PatFirstName, PatLastName, Address, PhoneNumber FROM patient WHERE
PhoneNumber = 'NULL';
```

-- Lists patient first name in the patient table for those patients whose phone number is null. Instead of sending an SMS as there isn’t a number on the patients record, Helen instead posts home a reminder about an appointment booked.

Rule 4: Dynamic relational online catalogue

Database dictionary (catalogue) must have description of Database. The database must contain certain system tables whose columns describe the structure of the database itself such as table_schema, table_type etc.

MySQL

SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'project';

Rule 5: Comprehensive data sub-language Rule

SQL supports a dentist like Mary Mulcahy to access the records of patient/ payment tables/views (SELECT) and manipulating the records by insert/delete/update. This rule also provides security by giving different levels of access rights to the staff who work at the clinic (GRANT and REVOKE). Mary for example can GRANT and REVOKE privileges on various database objects in the SQL Server.

MySQL

GRANT SELECT, INSERT, UPDATE, DELETE ON patients TO Mary;

--Grants Mary the privilege to select tables/views and update/delete patient/treatment records in the project database.

CREATE VIEW records_1 AS SELECT patient.PatFirstName, patient.PatLastName,
appointment.AppNumber, appointment.Date, appointment.DateOfStartOfTreatment,
appointment.DateOfEndOfTreatment, appointment.DentistNumber, appointment.PatNumber
FROM patient INNER JOIN appointment ON patient.PatNumber = appointment.PatNumber;

SELECT * FROM records_1;

--Helen can create a view called records_1 with PatFirstName, PatLastName for patient and AppNumber, Date, DateOfStartOfTreament, DateOfEndOfTreament, DentistNumber for appointment for those patients who have appointments next week.

Rule 6: View updating rule

We can use this rule when we want to create view to merge columns from different tables such as patient and payment. The following SQL statement creates a view called records_2 with the PatFirstName, PatLastName for Patient and PaymentNumber, PaymentDesc, OutstandingBalance for payment.

MySQL

CREATE VIEW records_2 AS SELECT patient.PatFirstName, patient.PatLastName,
payment.PaymentNumber, payment.PaymentDesc, payment.OutstandingBalance FROM
patient INNER JOIN payment ON patient.PatNumber = payment.PatNumber;

```
SELECT * FROM records_2;
```

--Mary Mulcahy creates a view for the treatment fees guideline book which selects all rows from both the patient and payment table as long as there is a match between the columns.

Rule 7: Relational level operation rule

A relational database must have insert, update, and delete operations. The dental clinic can be busy and therefore data is increasing day by day and we are inserting patient records in the database through insert command. To display we use select command and to update data we use update command. So these commands are executed very often in databases.

MySQL

```
UPDATE records_2 SET OutstandingBalance = 150.00 WHERE PaymentNumber= 'PA03';
```

--Mary from time to time updates the treatment fee guideline book in views.

Rule 8: Physical Data Independence

For example, relations in the dental clinic database are stored as unordered files. If the change from unordered files to some other representation would not require a change in the table design, attribute types etc.

Rule 9: Logical Data Independence

If User A: Helen adds a new column ‘cost’ in her view/table it will not affect the external view user, User B: Mary, but the internal view of the database has been changed for both users. Now Mary can also print the cost.

For the table with the schema treatment(ProcedureNumber, ProcedureDate, description, cost, PatNumber), a view with the attributes (ProcedureNumber, ProcedureDate) will not be affected if any other attributes of treatment is changed.

Rule 10: Integrity Independence

1. Entity integrity: no records can have NULL values in its Primary Key attribute. Helen can create a new table for patient records.

MySQL

```
CREATE TABLE patient_1(PatNumber Varchar(11) PRIMARY KEY, PatFirstName
VARCHAR(11), PatLastName VARCHAR(11), PhoneNumber VARCHAR(11),Address
VARCHAR(50), City VARCHAR(50), County VARCHAR(20), DateOfBirth DATE);

INSERT INTO `patient_1` (`PatNumber`, `PatFirstName`, `PatLastName`, `PhoneNumber`,
`Address`, `City`, `County`, `DateOfBirth`) VALUES ('P01', 'Carine', 'Schmitt', '0862368486', '345
View Ridge', 'Mallow', 'Cork', '1978-04-04');
```

--In this definition **PRIMARY KEY** means UNIQUE + NOT NULL, and we have defined PRIMARY KEY PatNumber as part of table definition itself using SQL.

Rule 11: Distribution Independence

A database should work properly regardless of its distribution across a network. This lays foundation of distributed database.

Rule 12: Non subversion rule

If low level access is allowed to a system it should not be able to subvert or bypass integrity rule to change data. This can be achieved by some sort of looking or encryption.

Example: violations of this rule: SQL Server 2008 infringes this through Bulk copy and disabling constraints and triggers options.