

Name: Aron Nutley

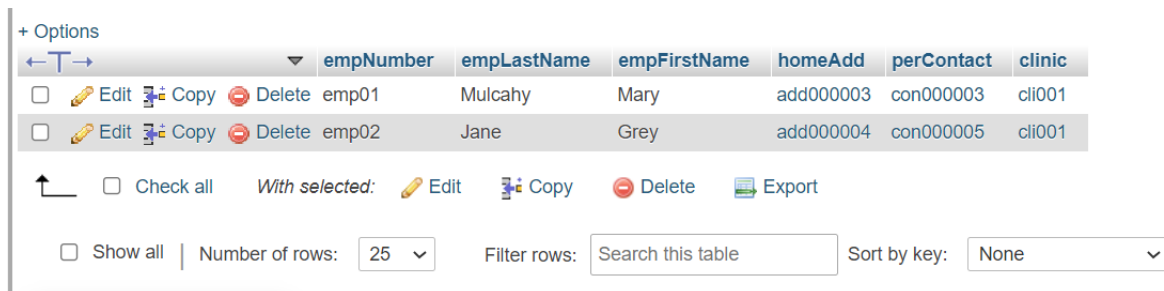
Student Number: G00388055

CODD's RULES

Rule 01 – The Information Rule

All information in the relational database is represented in exactly one and only one way by value of tables. SQL conforms with this rule as it not possible to store information in anything other than a table.

SQL Example: `SELECT * FROM `employee``



	empNumber	empLastName	empFirstName	homeAdd	perContact	clinic
<input type="checkbox"/>	emp01	Mulcahy	Mary	add000003	con000003	cli001
<input type="checkbox"/>	emp02	Jane	Grey	add000004	con000005	cli001

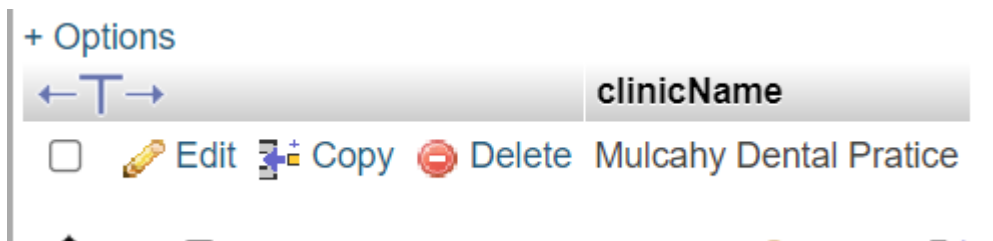
Fig 1: Result of above SQL Code when using project database

Rule 02 – Guaranteed Access Rule

Each and every datum is guaranteed to be logically accessible by resorting to a combination of table name, primary key value, and column name. Every scaler value in the DB must be logically addressable by specifying the name of the containing table, the name of the containing column and the primary key value of the containing row.

SQL Example:

`SELECT clinicName FROM `dental_practice`.`clinic` WHERE `clinicNumber` = 'cli001'`



clinicName
Mulcahy Dental Prattice

Fig 2: Result of above SQL Code when using project database

Rule 03 – Systematic Treatment of Null Values

The DB must allow each field to remain null. It must represent missing information and inapplicable information in a way that is systematic and distinct from all regular values.

SQL Example:

```
UPDATE `employee` SET `homeAdd` = NULL WHERE `employee`.`empNumber` = 'emp01';
```

+ Options		empNumber	empLastName	empFirstName	homeAdd	perContact	clinic
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		emp01	Mulcahy	Mary	add000003	con000003	cli001

Fig 3 – Before Code Execution

+ Options		empNumber	empLastName	empFirstName	homeAdd	perContact	clinic
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		emp01	Mulcahy	Mary	NULL	con000003	cli001

Fig 4 – After Code Execution

Rule 04 – Active Online Catalog

Database dictionary is the structure description of the complete database and it must be stored online. The catalog must be governed by the same rules as the rest of the database. Same query language on catalog as the database.

SQL Example:

```
SELECT * FROM information_schema.tables
```

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE	ENGINE	VERSION	ROW_FORMAT	TABLE_ROWS	AVG_ROW_LEN
def	information_schema	ALL_PLUGINS	SYSTEM VIEW	Aria	11	Page	NULL	
def	information_schema	APPLICABLE_ROLES	SYSTEM VIEW	MEMORY	11	Fixed	NULL	9
def	information_schema	CHARACTER_SETS	SYSTEM VIEW	MEMORY	11	Fixed	NULL	3
def	information_schema	CHECK_CONSTRAINTS	SYSTEM VIEW	MEMORY	11	Fixed	NULL	23
def	information_schema	COLLATIONS	SYSTEM VIEW	MEMORY	11	Fixed	NULL	2
def	information_schema	COLLATION_CHARACTER_SET_APPLICABILITY	SYSTEM VIEW	MEMORY	11	Fixed	NULL	1
def	information_schema	COLUMNS	SYSTEM VIEW	Aria	11	Page	NULL	
def	information_schema	COLUMN_PRIVILEGES	SYSTEM VIEW	MEMORY	11	Fixed	NULL	28
def	information_schema	ENABLED_ROLES	SYSTEM VIEW	MEMORY	11	Fixed	NULL	3
def	information_schema	ENGINES	SYSTEM VIEW	MEMORY	11	Fixed	NULL	7
def	information_schema	EVENTS	SYSTEM VIEW	Aria	11	Page	NULL	
def	information_schema	FILES	SYSTEM VIEW	MEMORY	11	Fixed	NULL	40
def	information_schema	GLOBAL_STATUS	SYSTEM VIEW	MEMORY	11	Fixed	NULL	63
def	information_schema	GLOBAL_VARIABLES	SYSTEM VIEW	MEMORY	11	Fixed	NULL	63

Fig 5 -Resulting Table from SQL Code Above

Rule 05 – Comprehensive Data Sublanguage Rule

This rule mandates the existence of a relational database language to manipulate data. SQL is not specifically required. The language must provide all manners of access to the data stored in the database.

SQL Example 01:

```
SELECT * FROM `dental_practice`.`employee` WHERE `empNumber` = 'emp01'
```

+ Options		empNumber	empLastName	empFirstName	homeAdd	perContact	clinic
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete		emp01	Mulcahy	Mary	NULL	con000003	cli001

Fig 6 – Result of Example 01

SQL Example 02:

```
SELECT empFirstName FROM `dental_practice`.`employee` WHERE `empNumber` = 'emp01'
```

				empFirstName
<input type="checkbox"/>				Mary

Fig 7 Result of Example 02

Rule 06 – View Updating Rule

All views that are theoretically updatable are also updatable by the system.

Options

				addressLine1	addressLine2	county			
<input type="checkbox"/>		Edit		Copy		Delete	4 Kilmoney Rd Lower	Kilmoney	Cork
<input type="checkbox"/>		Edit		Copy		Delete	Western Rd	Mardyke	Cork
<input type="checkbox"/>		Edit		Copy		Delete	9 Bluebell Lane	Forest Hill	Cork
<input type="checkbox"/>		Edit		Copy		Delete	10 Bluebell Lane	Forest Hill	Cork
<input type="checkbox"/>		Edit		Copy		Delete	112 Brightwater	NULL	Cork
<input type="checkbox"/>		Edit		Copy		Delete	54 Brightwater	NULL	Cork
<input type="checkbox"/>		Edit		Copy		Delete	172 Harty's Quay	Rochestown Rd	Cork
<input type="checkbox"/>		Edit		Copy		Delete	171 Harty's Quay	Rochestown Rd	Cork
<input type="checkbox"/>		Edit		Copy		Delete	43 Rovers Row	NULL	Cork
<input type="checkbox"/>		Edit		Copy		Delete	99 Ref Baloon Warf	NULL	Cork
<input type="checkbox"/>		Edit		Copy		Delete	44 Bluebell Lane	NULL	Cork
<input type="checkbox"/>		Edit		Copy		Delete	123 Fake St	Patricks Hill	Cork
<input type="checkbox"/>		Edit		Copy		Delete	18 Calm Meddows	NULL	Cork

Fig 8 Show View Called test_01

SQL Example:

```
CREATE OR REPLACE VIEW test_01 AS SELECT addressLine1, addressLine2 FROM address
```

Options				addressLine1	addressLine2
<input type="checkbox"/>				4 Kilmoney Rd Lower	Kilmoney
<input type="checkbox"/>				Western Rd	Mardyke
<input type="checkbox"/>				9 Bluebell Lane	Forest Hill
<input type="checkbox"/>				10 Bluebell Lane	Forest Hill
<input type="checkbox"/>				112 Brightwater	NULL
<input type="checkbox"/>				54 Brightwater	NULL
<input type="checkbox"/>				172 Harty's Quay	Rochestown Rd
<input type="checkbox"/>				171 Harty's Quay	Rochestown Rd
<input type="checkbox"/>				43 Rovers Row	NULL
<input type="checkbox"/>				99 Ref Baloon Warf	NULL
<input type="checkbox"/>				44 Bluebell Lane	NULL
<input type="checkbox"/>				123 Fake St	Patricks Hill
<input type="checkbox"/>				18 Calm Meddows	NULL

Fig 9 – Updated View

Rule 07 – High Level Insert, Update and Delete

The system must support set-at-a-time insert, update, and delete operators. This means that data can be retrieved from a relational database in sets constructed of data for multiple rows and/or multiple table.

SQL Example:

```
INSERT INTO `patient` (`patNumber`, `parLastName`, `patFirstName`, `personalDetails`, `contact`, `address`) VALUES ('pat0027', 'McKenna', 'David', 'per0023', 'con000019', 'add000016'), ('pat0028', 'Murphy', 'John', 'per0020', 'con000019', 'add000017');
```

<input type="checkbox"/>				pat0027	McKenna	David	per0023	con000019	add000016
<input type="checkbox"/>				pat0028	Murphy	John	per0020	con000019	add000017

Fig 10 – SQL Example for Rule 07

Rule 08 – Physical Data Independence

Changes to the physical level must not require a change to an application based on the structure. The applications must still work using the same syntax, even when changes are made.

Rule 09 – Logical Data Independence

If there is change in the logical structure of the database the user view of data should not change. An example of this is if a table was split into two, a new view should give the result of the two tables joined.

Rule 10 – Integrity Independence

Integrity constraints must be specified separately from application programs and stored in the catalog. It must be possible to change such constraints as and when appropriate without unnecessarily affecting existing applications.

Rule 11 – Distribution Independence

A database should work properly regardless of its distribution across a network. Even if the DB is physically distributed, with stored data in pieces, the end user should get an impression that it is stored at the same place.

Rule 12 – Non-Subversion Rule

This rule requires that alternate methods of accessing the data are not able to bypass integrity constraints, which means that users can't violate the rules of the database in anyway. This rule is followed for most SQL servers as getting to the raw data and changing values other than by the methods prescribed by the DB.

References:

Ref 01: <https://link.springer.com/content/pdf/bbm%3A978-1-4302-0867-9%2F1.pdf>

Ref 02: <https://www.w3resource.com/sql/sql-basic/codd-12-rule-relation.php>

Ref 03: <https://www.studytonight.com/dbms/codd-rule.php>

Ref 04: <https://www.essentialsql.com/what-is-a-data-dictionary/>