SQL DRILLS #2  
Due: Dec. 10th, 2020

**#1. CARTESIAN JOINS**

Part 1: 25 \* 10 = 250

Part 2:

|  |  |
| --- | --- |
| 1 | 10 |
| 1 | 11 |
| 1 | 12 |
| 2 | 10 |
| 2 | 11 |
| 2 | 12 |
| 3 | 10 |
| 3 | 11 |
| 3 | 12 |
| 4 | 10 |
| 4 | 11 |
| 4 | 12 |

**#2. FOREIGN KEYS**

Employees: employee\_id (primary key), first\_name, last\_name, department\_id (foreign key)  
Deprartments: id (primary key), dept\_name

**#3. ACID**

Atomicity: This property in SQL means that either all of the operations inside a transaction take place or none of them – the transaction acts on several pieces of information ONLY if all pieces successfully save. This transaction has an “all or nothing” application. Insertions, updates, and deletions are all examples of operations that need to successfully complete for the entire transaction to run.

Consistency: This property in SQL ensures the consistency throughout the database, meaning no matter what happens in the middle of a transaction, the transaction will not be left incomplete. If the transaction completes successfully, then it will apply all the changes to the database. If there is an error in the transaction, it will roll back all the changes already made, and the database will be restored to the state it was in before the transaction.

Isolation: This property makes sure that every single transaction is individual, therefore, one transaction can’t access the result of other transactions until the previous transaction is complete. One can also not perform the same operation using multiple transactions at the same time.

Durability: Once the transaction is complete, then the changes it has made to the database will be permanent. This property helps to safeguard all our committed data and protects us from unusual changes, as well as any system crashes.

**#4. CASE**

CASE  
 WHEN species = ‘duck’ THEN ‘mouse’  
 WHEN species = ‘mouse’ THEN ‘duck’  
END

**#5. INDEX**

Part 1: An index in SQL is an on-disk structure associated with a table or view that speeds up retrieval of rows from the specified table or view. An index contains keys built from one or more columns in the table or view. These keys are then stored in a structure that enables SQL to find the rows or rows associated with the key values quickly and efficiently.

Part2:

* Clustered index
* Nonclustered unique index
* XML index
* Spatial index
* Clustered columnstore index
* Nonclustered columnstore index
* Nonclustered hash index