**01-Cartesian\_Joins**:

Part 1:

* 35

Part 2:

* This column will have 2 columns, both named “ID”. The first column will have values 1, 2, 3, 4, 5, and the second column will have values 10, 11, and 12.

**02-Foreign\_Keys:**

Part 1:

Employees:

|  |  |  |  |
| --- | --- | --- | --- |
| Employee\_id | First\_name | Last\_name | Department\_id |
| 14 | Jan | Jarsson | 45 |
| 17 | Sam | Samuels | 45 |

Departmnents:

|  |  |
| --- | --- |
| Id | Dept\_name |
| 45 | webdev |
| 45 | Webdev |

**03-ACID:**

Part 1:

* Atomicity: either all operations take place, or none of the operations take place. The operations consist of insert, update, and delete.
* Consistency: the database cannot be ran in a half-completed state. Either the code is correct, and all statements run. Or the code is incorrect somewhere, so nothing runs and error msgs are produces.
* Isolation: each transaction is individual, and transactions that use outputs of previous transactions won’t be able to run until all the relevant transactions completed.
* Durability: once a transaction is completed, the changes are permanent until specifically changed

**04-Case:**

Part 1:

SELECT \*

CASE

WHEN animal\_name = ‘Mickey Mouse’

THEN species = 'mouse'

WHEN animal\_name = ‘Donald Duck’

THEN species = duck

END duration

FROM table\_name

**05-Index:**

Part 1:

* An index is like a key, or pointer to data in a table. An index will hold certain data that can be referenced or called later in code

Part 2:

* Clustered: store data or rows based on fundamental value, such as using a case statement to set that value
* Non-clustered: each value pale has a pointer to the data row that contains the relevant information
* Unique: doesn’t contain any duplicates
* Filtered: when a column has only a small number of relevant values for queries on the subset of values
* Column store: stores data in a column-based format
* Hash: A range of N buckets or slots containing a pointer and a row on each bucket or slot. Used the Hash function F(K, N)