SQL – Structured Query Language.

Se numeste asa, deoarece foloseste interogari structurate pentru a adauga si obtine date din tabele.

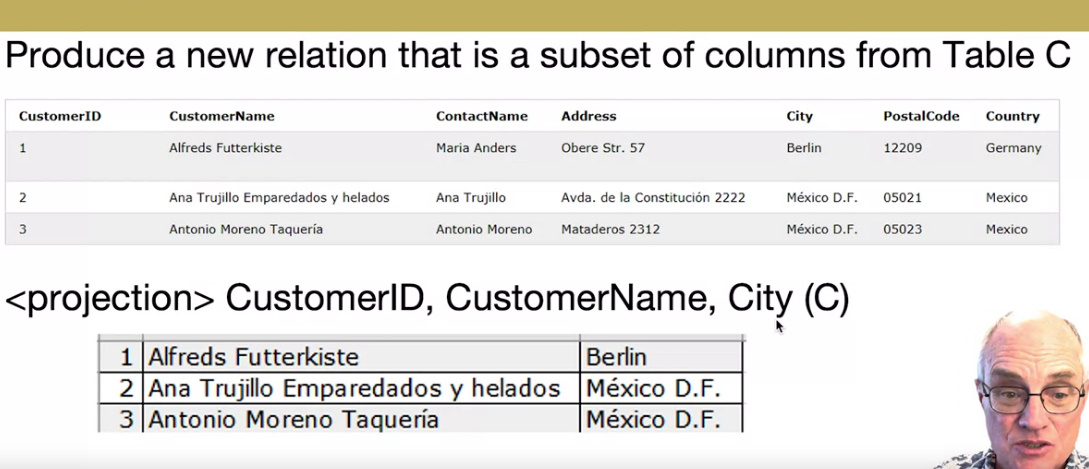
**Dice.com**

**Algebra Relationala**

* Algebra Relationara se bazeaza pe operatii asupra multimilor
* O multime este o relatie
* Variabilele sunt relatii in ALGEBRA relationara. De ex: 5x + 7y, x si y sunt variabile si deci relatii
* Operatii:
* Uniunea
* Intersectia
* Diferenta
* Selectia
* Proiectia
* Produsul
* Join

**Proiectia**

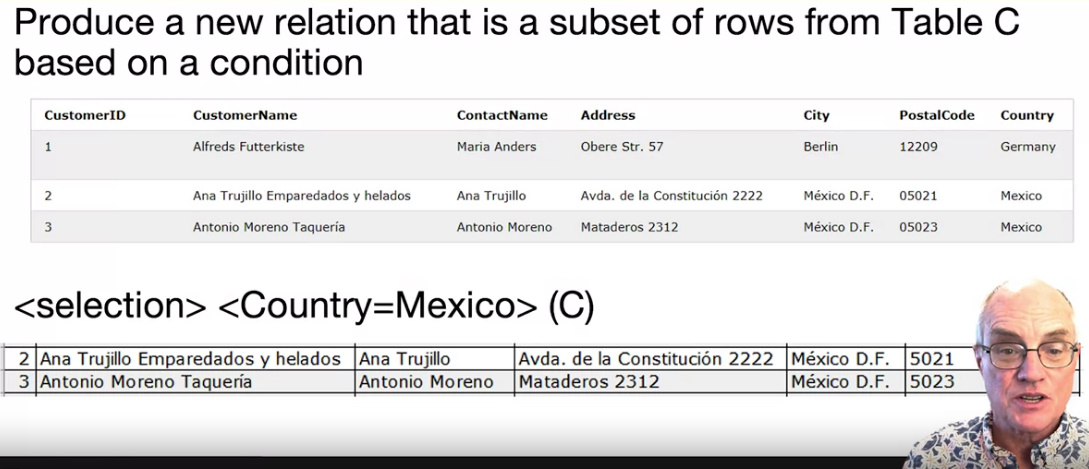
Produce o noua relatie care este o submultime de **coloane** din relatia originala.



<project> CustomerID, CustomerName, City (C) – din relatia C extragem submultimea formata din toate randurile coloanelor CustomerID, CustomerName si City

**Selectia**

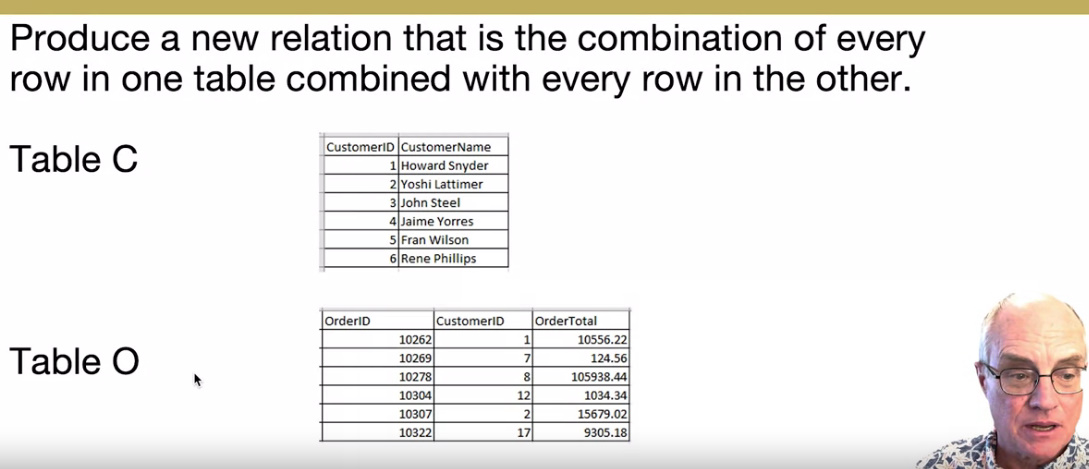
Produce o noua relatie care este o submultime de **randuri** din relatia originala.



Selectia poate avea si conditii, cum mai sus **<Country=Mexico>**

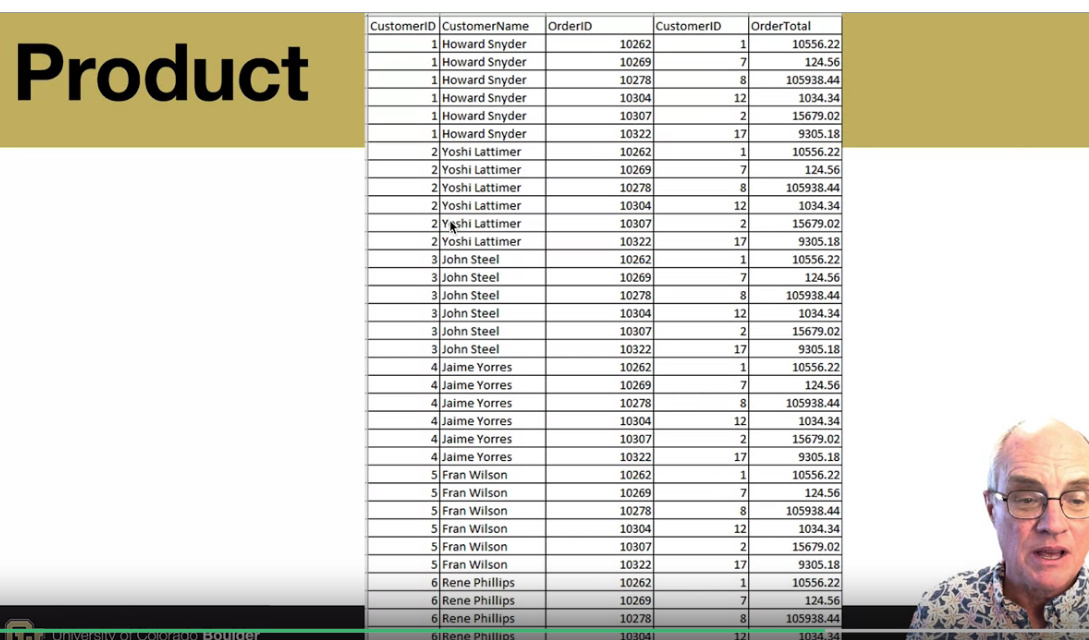
**Produsul**

Produce o noua relatie care este combinarea fiecarui rand dintr-un tabel cu randurile din alt tabel.



Coloanele ambelor tabele se lipesc, si formeaza un singur tabel cu coloanele din ambele tabele, si daca sunt coloane identice, ele se vor repeta oricum.

C combinat cu O:

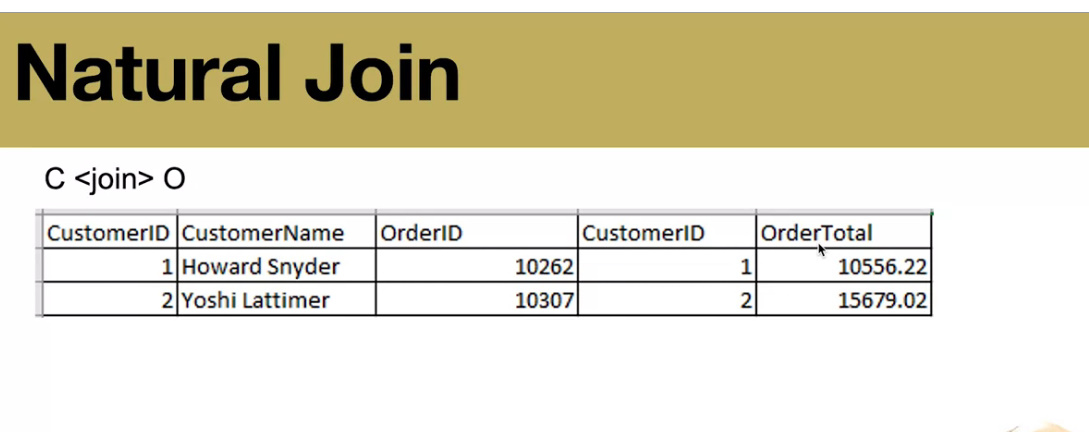




Vedem ca sunt randuri care au atribute cu valori identice. De ex, prima linie are CustomerID identic, dar restul iese ca nu au nicio legatura cu acest Customer, ele doar se combina inutil cu el.

**Jonctiunea(Join) Naturala**

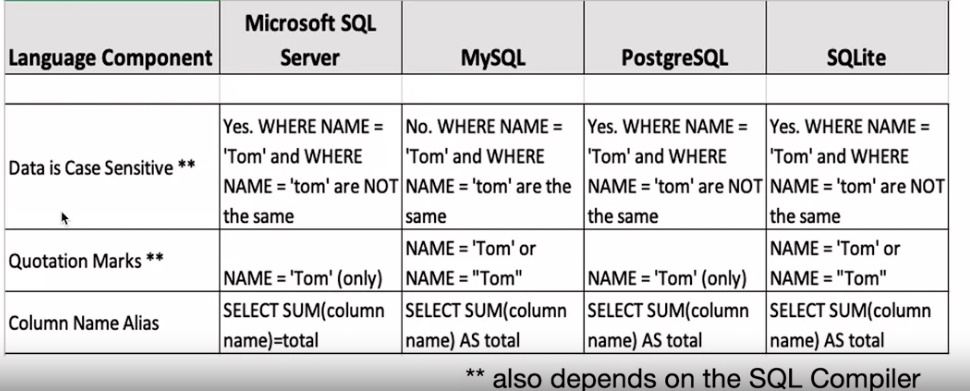
Combina fiecare linie dintr-un tabel cu toate liniile din alt tabel, dar verificand ca valorile unui anumit atribut sa fie egale. Asa se rezolva repetarea de la Produs de mai sus.



Aici s-au combinat doar liniile ce au CustomerID identic.

**SQL Standarts**

* Unele companii au creat propria implementare de SQL, pentru a-l imbnatati.
* De ex, cei care au creat MySQL, Microsoft SQL sau PostgreSQL au implemntat putin diferit SQL unele de altele
* Fiecare din ele foloseste propriile SQL commands specifice.

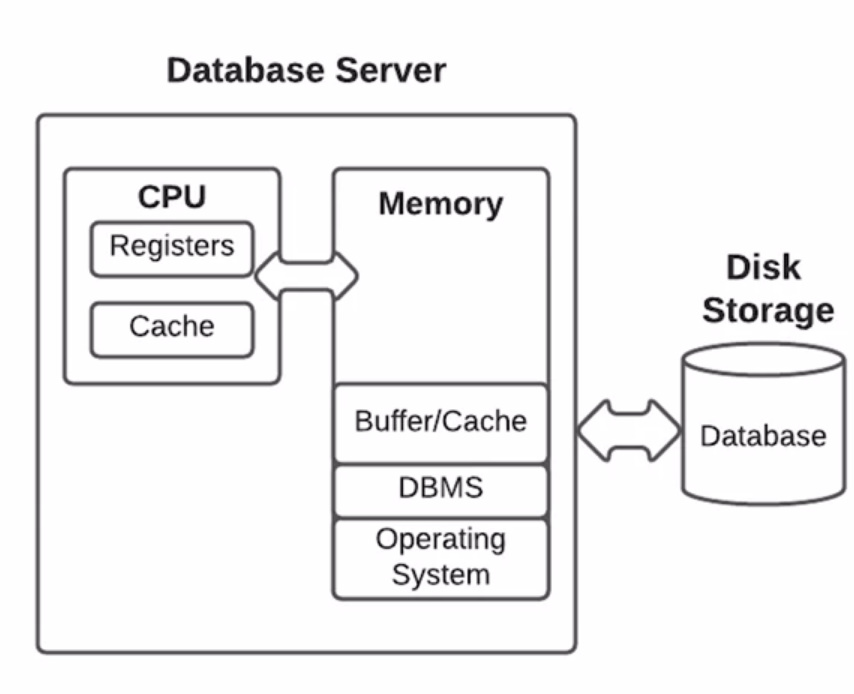


**Interogarile**

* SQL nu este chiar un limbaj de programare, ci un Query Language
* Nu este procedural. El nu executa cate o instructiune rand pe rand din secventa la un moment dat, ci executa toata interogarea deodata. Deci nu executa linie cu linie, ci totul deodata, intreaga interogare, nu ca limbajele de programare
* Toate inputurile si outputurile sunt tabele
* Tabelul de output e numit Answer Set
* Unele interogari pot produce interim(temporary) answer sets. De ex, cand cream subinterogari si facem select pentru a avea un tabel temporar pe baza la care vom face alt select. Nici nu il vedem macar.
* Cand cream o interogare complexa, cu sortari si selectari, SQL creaza diferite tabele, ce noi nu le vedem, si le tot modifica si recreaza, asa incat noi vedem la final tabelul necesar.
* SQL nu poate pur si simplu fi compilat si rulat pe PC ca Java sau C++. Avem nevoie de DBMS
* SQL nu poate crea fisiere .exe

**Executarea unei interogari**

1. SQL verifica codul sintactic
2. Leaga tabelele si coloanele la care ne referim in interogare, adica le colecteaza si le leaga
3. Creaza un executable file, adica creaza codul nostru in cod masina si il trimite la optimizer
4. Optimizer creaza si un plan de executie, pentru a executa query cand mai rapid si eficient posibil
5. DBMS executa planul de executie



**CPU:**

CPU e cel care executa instructiunile trimise.

El face miliarde de operatii pe secunda, la nivel de **nanosecunde**

**Registers** – rezultatele obtinute in urma executarii instructiunii sunt stocate aici

**Cache** – instructiunile de executat ce vin de la Memory sunt aici inregistrate. Deoarece Memory e mai lenta, nu poate tine pasul cu CPU, si ea trimite adresele de memorie de executat catre CPU si CPU le citeste in cache mai rapid.

**Memory**

Face milioane de operatii pe secunda, la nivel de **microsecunde**

Memory e mult mai lent ca CPU, si pentru a tine cu el ritmul, trimite la Cache adresele de memorie si CPU le citeste.

Memory incarca Operation System, apoi DBMS .

**Buffer/Cache** – zona speciala din memory folosita de database software pentru a pastra ce vine de la disc. Baza de date trebuie sa citeasca date din Disk, sa le pastreze,sa faca ceva cu ele si sa le trimita inapoi. Acestea sunt Inpout si Outpout operations, numite Disk IO.

**Disk**

E si mai lent ca Memory si face operatiile per **milisecunda**.

DataBaseServer poate face milioane de operatii in timpul care i-ar lua sa faca un IO (input sau output) la Disk

Query Tunning sau optimizatorul face tot posibilul sa reduca Disk IO.

Optimizatorul face tot posibilul pentru a executa cat mai rapid un query si a consuma cat mai putine resurse.

