* كل اسم يحوي معلومات مرشح أن يكون كيان باستثناء ERD.
* كل كيان له مجموعة واصفات:

1. بسيطة: مثل الاسم.
2. مشتقة: نحصل عليها من واصفات أخرى.
3. متعددة القيم: مرشحة لتكون كيان (لها أكثر من قيمة)، مثلاً User له أكثر من email.

SQL

تعد أكثر لغات الاستعلام التجارية انتشاراً، وهي تركيب من لغة الجبر العلاقاتي والحساب العلاقاتي.

أنواع العلاقات

تقسم العلاقات بين الجداول إلى ثلاث أقسام

One to Many

One to One

Many to Many

1. علاقة One to One:

مثال:

Person

Passport

كل شخص لديه جواز سفر واحد وكل جواز سفر هو عائد لشخص واحد.

1. علاقة One to Many:

نضع الـ FK من جهة الـ many

FK

Employee

Company

مثال:

كل موظف تابع لشركة واحدة ولكن الشركة لديها أكثر من موظف.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| … | Emp3 | Emp2 | Emp1 | Name | ID |
| … | Null | … | … | MTN | 1 |
| Null | … | Null | … | Syriatel | 2 |
| … | … | … | … | … | … |
| … | … | … | … | … | … |

نضع الـFK عند الـ Employee وهذا يعني أنه يكفي وضع عمود واحد في جدول الـ Employee يشير إلى الـ Company. أما إذا وضعناه عند الـ Company سيصبح لدينا في جدول الـ Company لكل موظف (Employee) عمود وهذا مرفوض.

* ملاحظة: الفرق بين علاقة الـ One to One وعلاقة الـ One to Many هو أن الـ FK في علاقة الـ One to One لا يمكن تكراره ويجب وضع Constraint (قيمته لا تتكرر) أنه “unique”.

1. علاقة Many to Many:

Shop

Product

FK

FK

مثال:

المنتج يباع في أكثر من محل متجر والمتجر يحوي أكثر من منتج.

نكسر العلاقة

Product

Shop

|  |  |  |
| --- | --- | --- |
| Price | PID | SID |
|  |  |  |

جدول كسر العلاقة

ملاحظات

* الواصفات التي توضع في جدول كسر العلاقة هي واصفات خاصة بالعلاقة ما بين الكيانين.
* يوجد ضمن كل record رقم لكل entity حتى نستطيع التمييز بين الـ entities.
* Foreign Key “FK”: هو الذي يربط بين الجداول.
* Primary Key: هو نفسه الـ ID.
* عمود status يتم عليه الكثير من عمليات المقارنة لذلك لا يجب أن يكون string ويجب أن يكون لديه ID خاص.

لدينا الـ Database التالية كمثال:

Company

Employee

|  |  |  |  |
| --- | --- | --- | --- |
| ... | ... | name | ID |
| ... | ... | MTN | 1 |
| ... | ... | Syriatel | 2 |
| ... | ... | ... | 3 |
| ... | ... | ... | 4 |
| ... | ... | ... | 5 |
| ... | ... | ... | ... |

|  |  |  |  |
| --- | --- | --- | --- |
| age | address | name | ID |
| 20 | ... | Ahmad | 1 |
| 25 | ... | Hasan | 2 |
| 30 | ... | Khaled | 3 |
| 25 | ... | Wael | 4 |
| 22 | ... | Omar | 5 |
| ... | ... | ... | ... |

|  |  |  |
| --- | --- | --- |
| Salary | CID | EID |
| … | … | … |

Work

إن الواصفة salary لا تنتمي إلى الـ company لوحدها ولا إلى الـ Employee لوحده.

Email

|  |  |  |
| --- | --- | --- |
| EID | Email | ID |
| … | … | 1 |
| … | … | 2 |
| … | … | 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| … | Email3 | Email2 | Email1 |
| … | … | … | … |

إنشاء قاعدة معطيات Database (لغة تعريف المعطيات DDL)

باستخدام برنامج SQL-Server، وبعد أن قمنا بإنشاء مخططات الـ ERD وفهمها أصبح بإمكاننا إنشاء قاعدة معطيات:

🡺 New Database 🡺 Create Table

أعمدة + علاقات

* Create table Employee:

ID int, Name varchar (50), Address varchar (50), Age int, CID int, Primary Key (ID), Foreign Key (CID) references company (ID), Birthdate date.

وهذا يعني اسم العمود ونوعه.

* Create table Company:

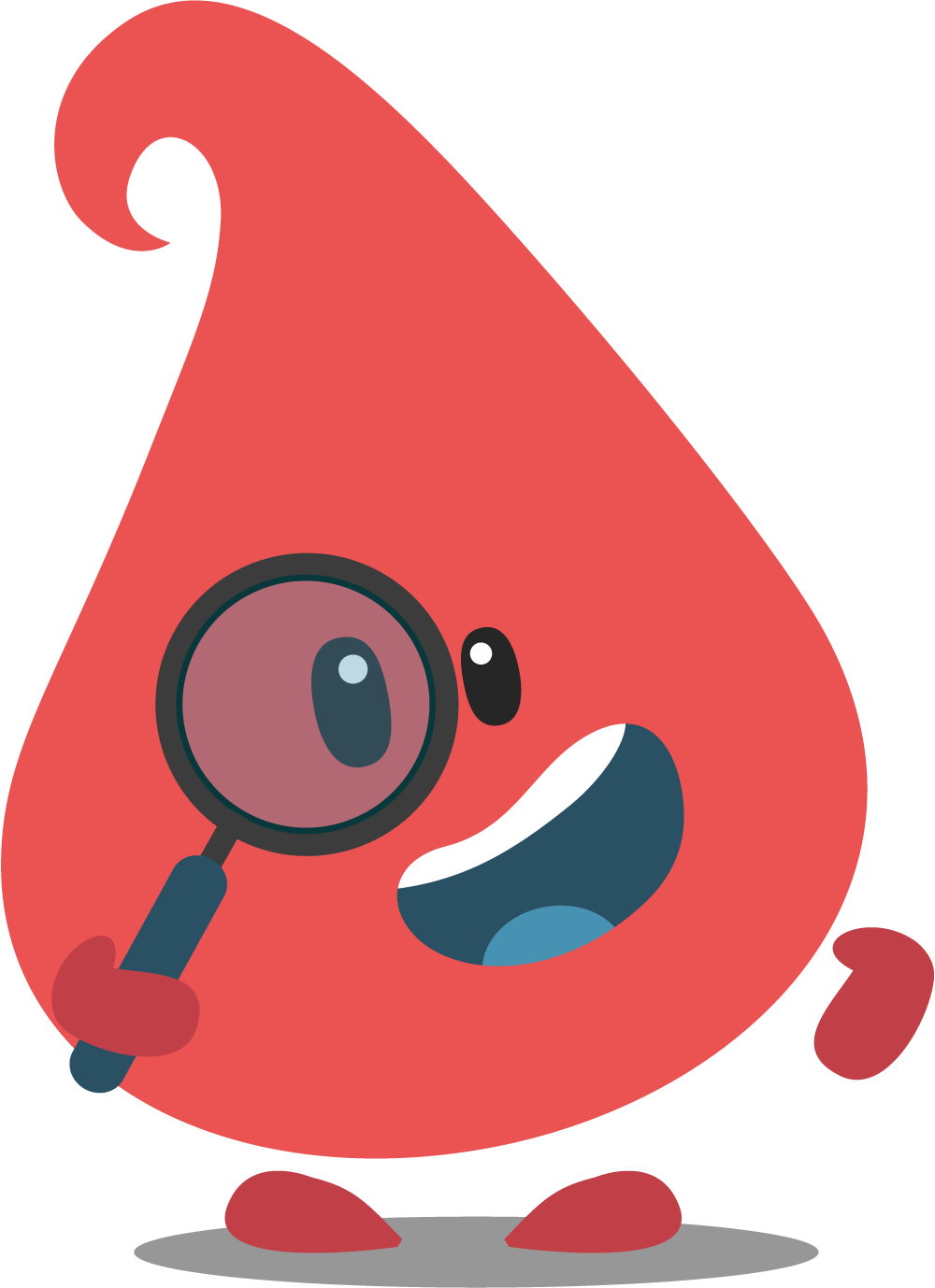
ID int, Name varchar(20), Primary Key (ID).

* تُعرّف شروط التكامل على المخطط العلاقاتي والمتضمنة:

عدم احتوائه على قيم غير معلومة not null.

تعريف المفتاح الرئيسي Primary Key.

تعريف قضية check (p) حيث p قضية.

* ملاحظة: تعريف المفتاح الرئيسي على واصفة يجعل اختبار عدم احتوائها على قيم غير معلومة آلياً.
* تعريف المجالات بلغة SQL:

يمكن تعريف أنماط مختلفة للمجالات بلغة SQL منها:

int: عدد صحيح.

number (p, d): رقم ممثل بالنقطة الثابتة.

real, double precision: رقم ممثل بالفاصلة العائمة مع دقة مضاعفة.

Date: تاريخ يحوي 4 خانات للسنة.

Time: الوقت في اليوم، ساعة، دقيقة، ثانية.

char (n): لتعريف سلسلة أحرف ذات طول ثابت حيث n طول ثابت يحدده المستثمر.

varchar (n): لتعريف سلسلة أحرف ذات طول متغير حيث n الطول الأعظمي الذي حدده المستثمر.

ملاحظات

* الواصفة المشتقة: هي واصفة يمكن استنتاجها من معلومات متوافرة .مثال:

لدينا واصفة Birthdate منها يمكننا استنتاج العمر age وجعله واصفة.

* يُسمح باستخدام القيم غير المعلومة في جميع أنماط المجالات، والتصريح يكون واصفة لا تقبل قيماً غير معلومة (يُحرّم استخدام القيم غير المعلومة لهذه الواصفة).
* يمكن للمستثمر أن يُعرف مجالات خاصة به باستخدام لغةSQL كالآتي:

create domain person-name char (20) not null

“Be nice to nerds. You may end up working for them. We all could.”

– Charles J. Sykes

Delete and Insert

Drop

تسمح تعليمة **drop table** بحذف جميع المعلومات المتعلقة بالعلاقة من قاعدة المعطيات.

drop table Employee;

ملاحظة: يجب حذف جدول الـ Employee أولاً ومن ثم جدول الـ company وليس العكس، أي يجب حذف العلاقة التي تربط ما بين الكيانين أولاً ثم نحذف الكيانين المرتبطين.

Insert into table

insert into employee

values ( I, "Ahmad", "Al mazzah" 20,1);

تستخدم لإدخال سطر جديد على Entity (الجدول).



سؤال؟

عندما نريد القيام بعملية **insert** للموظف جديد في الشركة، هل من المعقول أن نتذكر ID آخر موظف تم توظيفه حتى نتمكن من تحديد ID الموظف الجديد والقيام بـ عملية insert؟

بالطبع لا، ولهذا وُجِد ما يسمّى بـ **Identity** بدلاً من الـ ID.



بالعودة إلى create table Employee:

( ..., Identity ( 1 , 1 ) , ... )

رقم البداية

مقدار الزيادة

🡸 تصبح عملية الـ **insert** كالتالي:

insert into employee

values ("Ahmad", “Al mazzah”, 20,1);

أي استغنينا عن الـ ID لوجود الـ Identity التلقائية.

إدخال أعمدة محددة

insert into employee (Name, age, CID)

values ("Ahmad", 20, 1);

في هذه الحالة كل الأعمدة التي لم يتم تحديدها وإسناد قيم لها يتم وضع null فيها.

حذف سطر محدد

delete from employee

where ID=5;

* الفرق بين drop و delete:

drop تحذف الجدول مع البيانات كاملاً أمّا delete تحذف البيانات (الأسطر) أي لا يزال بإمكاننا القيام بعملية insert.

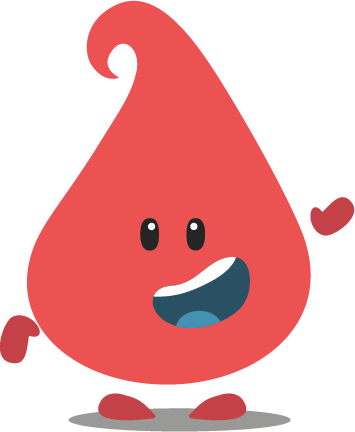
* ملاحظة هامة: عدم كتابة where condition أي كتابةdelete from employee; تعمل عملاً مشابهاً لـ drop table employee.

تعديل جدول بعد بنائه

تسمح تعليمة **alter table** بإضافة واصفات إلى مخطط علاقة موجودة، وبحيث تأخذ هذه الواصفة قيمة غير معلومة في جميع الحدوديات الموجودة سابقاً في العلاقة.

alter table Employee Add email varchar (50)

مثلاً: نملأ جدولاً يحوي 1000 موظف وقمنا باستخدام تعليمة **alter table...** يصبح لدينا عمود Emailلـ 1000 موظف بإسناد قيمته null. ولتعبئتهم يمكن استيراد ملف Excel كـ table في SQL Server.



“There is only one thing that makes a dream impossible to achieve: the fear of failure.”

– Paulo Coelho

**- The End -**