**Môi trường ứng dụng**

1. Cài đặt SQL Server
2. Cài đặt SSMS
3. Phân biệt SQL Server và SSMS
4. Dùng SSMS connect vào SQL Server

**Truy vấn**

1. Làm quen với một số câu truy vấn đơn giản

select 1

select 'Chào mừng bạn đến với khóa học Data Engineer'

select N'Chào mừng bạn đến với khóa học Data Engineer'

select @@VERSION

select getdate()

1. Đặt alias
2. Tạo database BikeStores và load dữ liệu
3. Kiểm tra số lượng bảng và số lượng dữ liệu (row) trong mỗi bảng trong database BikeStores
4. Tạo và view Database Diagrams cho database BikeStores
5. Thực hiện một số câu truy vấn đơn giản với bảng
6. Phân tích dữ liệu trước sau mỗi phare trong câu truy vấn (thứ tự thực hiện câu truy vấn)

SELECT

state

,city

,COUNT(\*) AS num\_customers

FROM

sales.customers a

WHERE

phone is not null

GROUP BY

state, city

HAVING

COUNT(\*) > 5

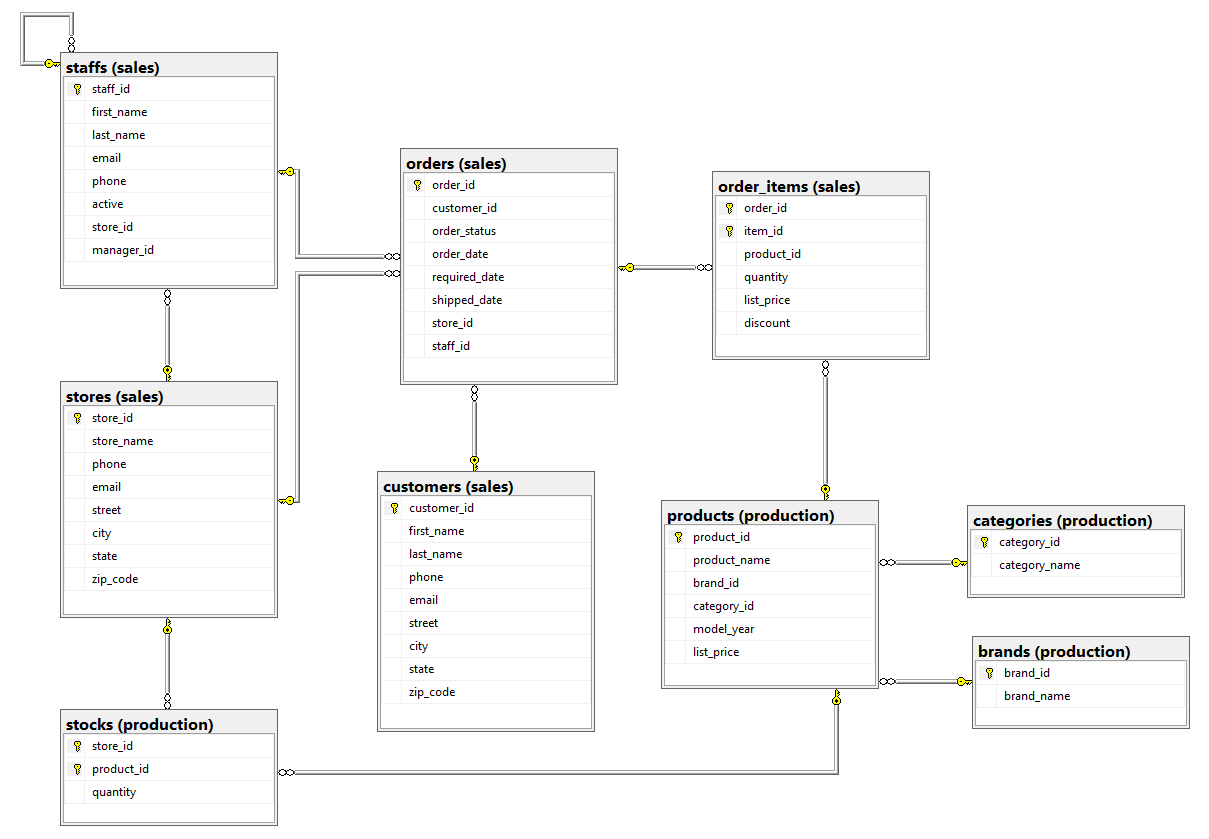
ORDER BY

state, city

1. Phân biệt giá trị NULL với các giá trị khác
2. Phân biệt COUNT(\*) và COUNT([column])
3. Lấy các dữ liệu khác nhau của 1 nhóm (GROUP BY)
4. Mô phỏng các loại JOIN

* Outer join
* Inner join
* Left join
* Union
* Cross join

1. Phân tích data model của database BikeStores



1. Lấy thông tin đầy đủ của một order

* Ngày tạo order, ngày chốt đơn, ngày ship hàng,
* Tên khách (full name), số đt khách, email khách, địa chỉ khách (street, city, state, zip code)
* Tên cửa hàng, số đt cửa hàng, email cửa hàng, địa chỉ cửa hàng (street, city, state, zip code)
* Tên nhân viên bán hàng (full name), email nhân viên, số đt nhân viên

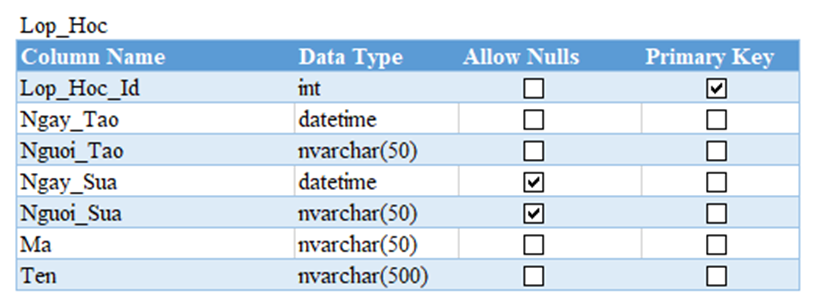
1. Lấy danh sách customer có số điện thoại
2. Lấy danh sách customer có số first\_name bắt đầu bằng chữ cái A
3. Thống kê Doanh số bán hàng theo từng sản phẩm
4. Thống kê số lượng đơn hàng của từng cửa hàng
5. Thống kê doanh số bán hàng của từng cửa hàng
6. Thống kê doanh số bán hàng theo từng nhân viên
7. Thống kê số lượng tồn kho theo từng sản phẩm
8. Thống kê doanh số bán hàng theo từng thương hiệu sản phẩm
9. Liệt kê danh sách sản phẩm chưa bán được cái nào
10. Liệt kê danh sách khách hàng chưa mua hàng lần nào
11. Liệt kê danh sách khách hàng đã mua 10 đơn hàng trở lên
12. Liệt kê khách hàng ở thành phố New York, bang NY
13. Thống kê doanh số bán hàng của từng cửa hàng theo năm
14. Thống kê sản phẩm có tồn kho lớn hơn 100 tại mỗi kho
15. Thống kê số khách hàng đã từng mua hàng tại cửa hàng Santa Cruz Bikes
16. Thống kê số khách hàng đã từng mua hàng tại cửa hàng Santa Cruz Bikes và địa chỉ ở thành phố Longview
17. Liệt kê sản phẩm có giá từ $500 đến $1000
18. Tính tổng doanh số bán hàng từng năm theo từng cửa hàng
19. Liệt kê danh sách khách hàng có first\_name kết thúc bằng chữ cái “u”
20. Liệt kê danh sách khách hàng có email của gmail
21. Trích xuất dữ liệu email domain khác nhau từ dữ liệu khách hàng (gợi ý: sử dụng các hàm *substring*, *charindex* và *len*)
22. Trích xuất đầu số điện thoại khác nhau từ dữ liệu khách hàng
23. Liệt kê danh sách thông tin nhân viên kèm thông tin người quản lý

**DML - Data Manipulation Language**

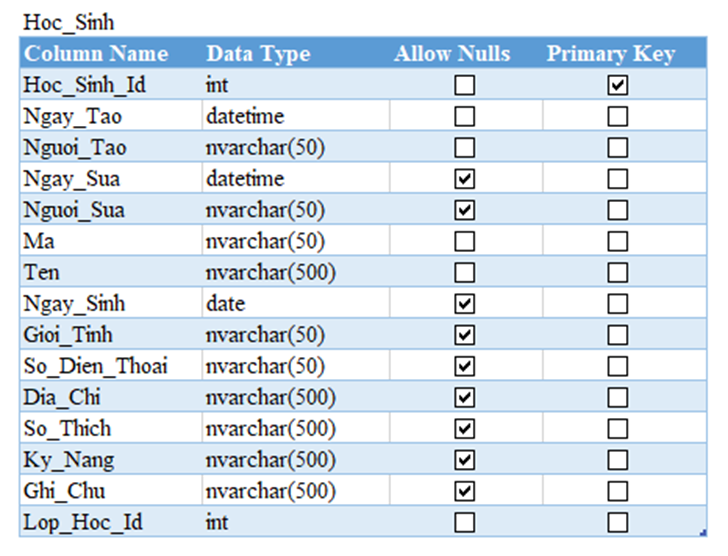
1. Tạo 10 row dữ liệu vào bảng khách hàng (sales.customer)
2. Tạo 3 row dữ liệu về nhãn hàng (production.brands)
3. Tạo 5 row dữ liệu sản phẩm
4. Tạo đơn hàng:
   * Khách hàng: Kimbery Nieves (id = 1259)
   * Cửa hàng: Rowlett Bikes
   * Nhân viên: Kali Vargas
   * Đơn hàng có 2 sản phẩm
     + Surly ECR 27.5 - 2018 (id=128), số lượng 4, giảm giá 10% và
     + Electra Townie Commute Go! Ladies' - 2018 (id 199), số lượng 6, giảm giá 15%;
5. Tạo đơn hàng:
   * Khách hàng: Cinda Rocha (id = 1307)
   * Cửa hàng: Baldwin Bikes
   * Nhân viên: Marcelene Boyer
   * Đơn hàng có 3 sản phẩm
     + Electra Cruiser Lux 7D - 2018 (id 224), số lượng 1, giảm giá 5%
     + Trek Farley Carbon Frameset - 2018 (id 123), số lượng 1, giảm giá 5%
     + Trek Powerfly 5 Women's - 2018 (id 202), số lượng 1, giảm giá 5%

**DDL - Data Definition Language**

1. Tạo bảng lưu dữ liệu lớp học có mô tả như trong hình



1. Tạo bảng lưu dữ liệu học sinh có cấu trúc như trong hình



1. Tạo 4 row dữ liệu cho bảng Lop\_Hoc
2. Tạo 30 row dữ liệu cho bảng học sinh (chia ra cho 4 lớp ở trên)
3. Thêm cột Ghi\_Chu vào bảng Lop\_Hoc, kiểu nvarchar(1000)

**Stored Procedure - Function**

1. Viết hàm tính tổng hai số a, b
2. Viết hàm tính tổng từ 1 đến n
3. Viết hàm tính tiền của một order\_items
   * Name: fcn\_order\_item\_payment
   * Input: order\_id, item\_id
   * Output: item price
4. Viết hàm tính tiền của một orders
   * Name: fcn\_order\_total\_payment
   * Input: order\_id
   * Output: order price
5. Viết hàm trả về thứ của ngày
   * Name: fcn\_DOW
   * Input: date
   * Output: thứ 2 → 2 … chủ nhật → 1
6. Sử dụng con trỏ in tên và thông tin của tất cả các sản phẩm
7. Duyệt qua các nhân viên. In tên nhân viên đó và tên quản lý tương ứng. Nếu không có quản lý ghi người đó là Quản lý cấp cao
8. **Viết procedure tạo đơn hàng**

-- viết procedure tạo đơn hàng:

-- sales.orders:

-- order\_id: tự tăng

-- customer\_id: lấy ngẫu nhiên trong bảng sales.customers

-- order\_status: mặc định 4

-- order\_date: cộng ngẫy nhiên 0 đến 30 ngày vào ngày hiện tại

-- required\_date: ngẫu nhiên trong khoảng order\_date + 20 ngày

-- shipped\_date: ngẫu nhiên trong khoảng required\_date + 15 ngày

-- store\_id: ngẫu nhiên trong bảng sales.stores

-- staff\_id: ngẫu nhiên trong bảng sales.staffs

-- sales.order\_items: số line random từ 3 đến 15

-- order\_id: order\_id đã tạo ở trên

-- item\_id: tăng dần theo số lượng item trong order này

-- product\_id: ngẫu nhiên trong bảng production.products

-- quantity: ngẫu nhiên trong khoảng 1 đến 12

-- list\_price: giá của sản phẩm đã chọn

-- discount: ngẫu nhiên trong các giá trị: 0, 5%, 10%, 15%, 20%

1. Viết thủ tục và hàm tìm giá trị max
   * Input: hai số a, b
   * Output: số lớn hơn trong a và b
2. Viết thủ tục và hàm trích xuất domain của 1 email
   * Input: email
   * Output: domain
   * Ví dụ: input: [abc123@hotmail.com](mailto:abc123@hotmail.com) → output hotmail.com
3. Viết thủ tục và hàm viết hoa chữ cái đầu của tên
   * Input: name
   * Output: tên viết hoa chữ cái đầu
   * Ví dụ: nguyễn văn abc → Nguyễn Văn Abc
4. Viết hàm tính tuổi
   * Input: ngày sinh
   * Output: tuổi
5. Viết hàm tính số order\_items của một order
   * Input: order\_id
   * Output: number\_of\_item
6. Viết hàm lấy order\_id mới nhất
7. Viết hàm tính tồn kho của một sản phẩm production.product

**View - Index - Transaction**

1. Tạo view thông tin khách hàng (bảng sales.customers) bao gồm các thông tin: customer\_id, full\_name, phone, email, address, state, zip\_code, số lượng đơn hàng đã mua, tổng tiền đã chi để mua hàng
2. Tạo view thông tin đơn hàng (sales.orders) bao gồm các thông tin:
   * Ngày tạo order, ngày chốt đơn, ngày ship hàng,
   * Tên khách (full name), số đt khách, email khách, địa chỉ khách (street, city, state, zip code)
   * Tên cửa hàng, số đt cửa hàng, email cửa hàng, địa chỉ cửa hàng (street, city, state, zip code)
   * Tên nhân viên bán hàng (full name), email nhân viên, số đt nhân viên
   * Giá trị của đơn hàng
   * Phân loại đơn hàng:
     + High: giá trị đơn hàng > 1500
     + Medium: giá trị đơn hàng < 1500, > 750
     + Low: giá trị đơn hàng < 750
3. Tạo non-clustered index trên cột first\_name, last\_name, phone, email trong bảng sales.customers
4. Tạo non-clustered index trên cột product\_name trong bảng production.products
5. Tái hiện, mô phỏng để hiểu rõ hơn về transaction
6. Tái hiện, mô phỏng để hiểu rõ hơn về lock dữ liệu
7. Tái hiện, mô phỏng 1 trường hợp deadlock
8. Import dữ liệu khách hàng vào bảng sales.customers

<https://docs.google.com/spreadsheets/d/1SvdGKQMSM_w0kV8HN_bcNjObxbMfiYeR/edit?usp=sharing&ouid=112859898253697715248&rtpof=true&sd=true>



**PROJECT**

Thiết kế database và tạo các procedure xử lý nghiệp vụ cho bài toán EVoucher

Chi tiết nghiệp vụ mô tả trong file SRS

<https://docs.google.com/document/d/1W8wdnXH0kht9PkLeUmXWBRuad9GOsU3N/edit?usp=sharing&ouid=112859898253697715248&rtpof=true&sd=true>

File dữ liệu Nhân viên

<https://docs.google.com/spreadsheets/d/1VgJPqQ65Jnu0BcbThK_C56tPoCdRBy6W/edit?usp=sharing&ouid=112859898253697715248&rtpof=true&sd=true>

File dữ liệu ngân sách

Ngân sách tháng 5/2022

<https://docs.google.com/spreadsheets/d/1VuX_kPh7Zh_vc6B49-MHAolXqWRNwIJu/edit?usp=sharing&ouid=112859898253697715248&rtpof=true&sd=true>

Ngân sách tháng 6/2022

<https://docs.google.com/spreadsheets/d/1Vgu6wAweteNnNHcFrCA8qwjFr3obhqY3/edit?usp=sharing&ouid=112859898253697715248&rtpof=true&sd=true>

File dữ liệu voucher

Voucher 100

<https://docs.google.com/spreadsheets/d/1Vf_s-i7NeGZ0CGuW4kZpVeE832M0R6ba/edit?usp=sharing&ouid=112859898253697715248&rtpof=true&sd=true>

Voucher 200

<https://docs.google.com/spreadsheets/d/1VetWqQQMBU54dB_OySPQcbG1h5dimo3U/edit?usp=sharing&ouid=112859898253697715248&rtpof=true&sd=true>

Voucher 500

<https://docs.google.com/spreadsheets/d/1VbhoZN9tbMG1H97cBi9stnLx-d2ZZpkH/edit?usp=sharing&ouid=112859898253697715248&rtpof=true&sd=true>

**Yêu cầu chi tiết Project**

1. Tạo database cho project
2. Thiết kế, tạo các bảng chức năng chung
   * App\_User
     + App\_User\_Id - nvarchar(50) - primary key
     + CreateUser - nvarchar(50) - not null
     + CreateDate - datetime - not null
     + UpdateUser - nvarchar(50)
     + UpdateDate - datetime
     + IsActive - bit
     + App\_Org\_Id - nvarchar(50) - not null - FK: App\_Org
     + UserName - nvarchar(50) - not null - unique
     + FullName - nvarchar(250) - not null
     + Email - nvarchar(250)
     + EmailConfirmed - bit
     + PhoneNumber - nvarchar(50)
     + PhoneNumberConfirmed - bit
     + AccessFailedCount - int
     + IsAdmin - bit
     + PasswordHash - nvarchar(250)
     + LastLogin - datetime
   * App\_Menu
     + App\_Menu\_Id - nvarchar(50) - primary key
     + CreateUser - nvarchar(50) - not null
     + CreateDate - datetime - not null
     + UpdateUser - nvarchar(50)
     + UpdateDate - datetime
     + IsActive - bit
     + App\_Org\_Id - nvarchar(50) - not null - FK: App\_Org
     + Name - nvarchar(250) - not null
     + TranslateKey - nvarchar(250)
     + Url - nvarchar(250)
     + Icon - nvarchar(50)
     + DisplayOrder - int
     + ParentId - nvarchar(50)
   * App\_Role
     + App\_Role\_Id - nvarchar(50) - primary key
     + CreateUser - nvarchar(50) - not null
     + CreateDate - datetime - not null
     + UpdateUser - nvarchar(50)
     + UpdateDate - datetime
     + IsActive - bit
     + App\_Org\_Id - nvarchar(50) - not null - FK: App\_Org
     + Code - nvarchar(50) - not null
     + Name - nvarchar(250) - not null
     + Description - nvarchar(1000)
   * App\_Org
     + App\_Org\_Id - nvarchar(50) - primary key
     + CreateUser - nvarchar(50) - not null
     + CreateDate - datetime - not null
     + UpdateUser - nvarchar(50)
     + UpdateDate - datetime
     + IsActive - bit
     + Code - nvarchar(50) - not null - unique
     + Name - nvarchar(250)
     + NameEn - nvarchar(250)
     + Type - nvarchar(50) - App\_Dic\_Domain: App\_Org.Type
     + Address - nvarchar(1000)
     + Description - nvarchar(1000)
     + ParentId - nvarchar(50)
     + DisplayOrder - int
   * App\_Setting
     + App\_Setting\_Id - nvarchar(50) - primary key
     + CreateUser - nvarchar(50) - not null
     + CreateDate - datetime - not null
     + UpdateUser - nvarchar(50)
     + UpdateDate - datetime
     + IsActive - bit
     + App\_Org\_Id - nvarchar(50) - not null - FK: App\_Org
     + Code - nvarchar(50) - not null
     + Value - nvarchar(50) - not null
     + Description - nvarchar(1000)
   * App\_File
     + App\_File\_Id - nvarchar(50) - primary key
     + CreateUser - nvarchar(50) - not null
     + CreateDate - datetime - not null
     + UpdateUser - nvarchar(50)
     + UpdateDate - datetime
     + IsActive - bit
     + App\_Org\_Id - nvarchar(50) - not null - FK: App\_Org
     + FilePath - nvarchar(550)
     + FileExt - nvarchar(50) - not null
     + FileName - nvarchar(250) - not null
     + FileSize - int - not null
     + FileContent - varbinary(max)
     + IsContentOnly - bit
     + IsTemp - bit
   * App\_Dic\_Domain
     + App\_Dic\_Domain\_Id - nvarchar(50) - primary key
     + CreateUser - nvarchar(50) - not null
     + CreateDate - datetime - not null
     + UpdateUser - nvarchar(50)
     + UpdateDate - datetime
     + IsActive - bit
     + App\_Org\_Id - nvarchar(50) - not null - FK: App\_Org
     + DomainCode - nvarchar(50) - not null
     + ItemCode - nvarchar(50) - not null
     + ItemValue - nvarchar(50) - not null
     + DisplayOrder - int
     + Description - nvarchar(1000)
   * App\_Sequence
     + App\_Sequence\_Id - nvarchar(50) - primary key
     + CreateUser - nvarchar(50) - not null
     + CreateDate - datetime - not null
     + UpdateUser - nvarchar(50)
     + UpdateDate - datetime
     + IsActive bit
     + App\_Org\_Id - nvarchar(50) - not null - FK: App\_Org
     + Code - nvarchar(50) - not null
     + Type - nvarchar(50)
     + Prefix - nvarchar(50)
     + Length - int
     + SeqValue - int
     + Description - nvarchar(1000)
   * App\_Log
     + App\_Log\_Id - nvarchar(50) - primary key
     + CreateUser - nvarchar(50) - not null
     + CreateDate - datetime - not null
     + App\_Org\_Id - nvarchar(50) - not null - FK: App\_Org
     + TableName - nvarchar(50)
     + RowId - nvarchar(50)
     + Action - nvarchar(50)
     + OldValue - nvarchar(1000)
     + NewValue - nvarchar(1000)
   * App\_Role\_Menu\_Ref
     + App\_Role\_Menu\_Ref\_Id - nvarchar(50) - primary key
     + CreatedDate - nvarchar(50) - not null
     + CreatedUser - datetime - not null
     + App\_Role\_Id - nvarchar(250) - not null
     + App\_Menu\_Id - nvarchar(250) - not null
   * App\_User\_Org\_Ref
     + App\_User\_Org\_Ref\_Id - nvarchar(50) - primary key
     + CreatedDate - nvarchar(50) - not null
     + CreatedUser - datetime - not null
     + App\_User\_Id - nvarchar(250) - not null
     + App\_Org\_Id - nvarchar(250) - not null
   * App\_User\_Role\_Ref
     + App\_User\_Role\_Ref\_Id - nvarchar(50) - primary key
     + CreatedDate - nvarchar(50) - not null
     + CreatedUser - datetime - not null
     + App\_User\_Id - nvarchar(250) - not null
     + App\_Role\_Id - nvarchar(250) - not null
3. Với mỗi bảng chức năng chung tạo các store nghiệp vụ thêm sửa xóa, get by id, get paging
4. Viết một số store nghiệp vụ riêng
   * Lấy dữ liệu menu tương ứng với App\_User
5. Thiết kế bảng dữ liệu nghiệp vụ EVoucher
   * Nhân viên
   * Danh mục loại voucher
   * Danh sách voucher
   * Quản lý ngân sách voucher
6. Với mỗi bảng chức năng nghiệp vụ EVoucher tạo các store nghiệp vụ thêm sửa xóa, get by id, get paging
7. Import dữ liệu nhân viên
8. Import dữ liệu ngân sách
9. Import dữ liệu voucher
10. Viết procedure thực hiện chức năng phân bổ voucher
11. Viết procedure thực hiện chức năng phân bổ mã voucher
12. Viết procedure thực hiện nghiệp vụ sử dụng voucher
13. Viết procedure báo cáo số lượng và giá trị voucher đã phân bổ theo tháng