**Check list**

0. Share github

- SQL query data from EAV table

- Query all product name for store English level

SELECT value

FROM `catalog\_product\_entity\_varchar` AS A, (

SELECT entity\_id

FROM `catalog\_product\_entity\_varchar`

WHERE attribute\_id = 73 and store\_id = 1)

AS B WHERE attribute\_id = 73

and store\_id = 0 and A.entity\_id <> B.entity\_id

UNION SELECT value

FROM `catalog\_product\_entity\_varchar`

WHERE attribute\_id = 73

and store\_id = 1

- Query all customer name, customer address

SELECT CONCAT(C1.firstname," ", C1.lastname) AS fullname, CONCAT(C2.city," ",C2.region) AS address FROM `customer\_entity` C1

LEFT JOIN customer\_address\_entity C2

ON C1.entity\_id = C2.parent\_id

- Observer practices

* Logs all front-end action (requests)
* Automatic discount 50% when the customer is logged in

- Plugin practices

- Inject $customer variable to block absolute\_footer with new PHTML template to show logged in user information

- Class preference example

- Argument replacement example

- Example of using repository class (ProductRepositoryInterface) to load a specific product from SKU

- Example of using factory class (ProductFactory) to load a specific product from SKU

- Example of using object manager directly to load a specific product from SKU

**1. When/where the layout loaded and parse in Magento?**

**2. How do Magento locate the controller/action to be excuted?**

step1: create routes.xml in etc/frontend or etc/adminhtml folder (first one is for frontend and second one is for admin controller).

step2: Add your custom configuration for controller in routes.xml, for example:

– router: id – standard(frontend)/admin

– route: id – your unique route id

– route: frontName – unique name in url, this is first part of url in base router (example/frontName/actionpath/actionclass/)

– module name – your module name

step3: Create your action class following the url structure above: Controller/Actionpath/Actionclass.php

**3. How do Magento process from friendly URL to controller/action ? Where is the code implementation?**

- The following process recognizes the home page empty path, and translates that to a data structure consisting of 3 elements: moduleFrontName, actionPath, actionName, which end up being 'cms', 'index', 'index', or 'cms\_index\_index' when composed as a page\_type id.

- Code implementation at: magento/framework/App/Router/Base.php.

**4. What does each table in Magento do? How does Magento store product, customers in databse ?**

- Magento 2 lưu trữ product, Customer :

Customer: bảng customer\_entity là bảng lưu trữ.

Bảng Customer\_entity là nơi lưu trữ thông tin chính về khách hàng.

Các bảng phụ :ustomer\_entity\_datetime, customer\_entity\_decimal, customer\_entity\_int, customer\_entity\_text, customer\_entity\_varchar chứa giá trị của các thuộc tính được phân loại theo loại dữ liệu tương ứng.

Catelog: bao gồm danh mục và sản phẩm.

Cách lưu trữ bảng danh mục tương tự như bảng của khách hàng, bảng chính là catalog\_category\_entity, các bảng phụ là các bảng có tiền tố như catalog\_category. Cần lưu ý từ bảng này rằng parent\_id là lưu trữ thông tin về mối quan hệ giữa các loại (loại chính).

Bảng catalog\_category\_product là để hiển thị mối quan hệ giữa sản phẩm và danh mục, sản phẩm nào thuộc loại nào.

Bảng có tiền tố catalog\_product\_entity giá trị lưu trữ của thuộc tính sản phẩm.

Bảng catalog\_product\_link thể hiện mối quan hệ giữa các sản phẩm, trong đó các loại mối quan hệ được lưu trữ trong catalog\_product\_link\_type, các giá trị của các thuộc tính mối quan hệ được lưu trữ trong bảng khác với tiền tố tương ứng.

Bảng catalog\_product\_option chứa thông tin về các tùy chọn mà sản phẩm có thể có. Bảng này được liên kết với catalog\_product\_bundle\_option

Bảng catalog\_product\_website hiển thị mối quan hệ giữa sản phẩm và trang web, sản phẩm trong Magento được tạo bởi các trang web riêng biệt.

**5. Purpose of each indexer?**

**-**Mục đích là để cải thiện performance bên ngoài trang storefront. Bởi vì kho database của magento2 là rất lớn, indexer sẽ giúp chuyển đổi các bảng EAV sang FLAT để perfomance khi tưonwg tác với trang frontend sẽ cao hơn.

**6. Purpose of flat tables ?**

**-** Giúp tăng tốc khi tương tác database bên trang storefront

**7.** **Observer and plugin comparison?**

**-** *observer* :

are capable of modifying the behavior of a Magento application because they are dynamically injected into the execution flow

- *plugin:*

is a class that modifies the behavior of public class functions by intercepting a function call and running code before, after, or around that function call.

**8. How to use an event like this $this->\_eventManager->dispatch($this->\_eventPrefix. '\_move\_before', $eventParams); ?**

Sử dụng Observer, ví dụ căn bản như sau:

b1: Tạo file Index.php theo directory app/code/Namespace/Module/Controller/Index/Index.php: nơi đây sẽ viết code dispatch như $this->\_eventManager->dispatch($this->\_eventPrefix. '\_move\_before', $eventParams);

B2: Tạo file events.xml theo directory app/code/Namespace/Module/etc/events.xml: nơi đây sẽ khai báo sự kiện

b3: Tạo file observer\_xxx.php theo directory app/code/Namespace/Module/Observer

**9. When do we have a conflict happen? And what happen when we have a conflict?**

- Conflict xảy ra khi 2 hay nhiều module cùng override 1 class nào đó.