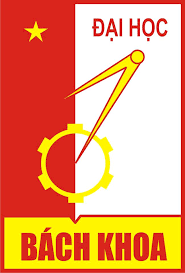
**※ TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI ※**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ**

**TRUYỀN THÔNG**

**-------------------------------------------------------------------------**



**-------------------------------------------------------------------------**

**Báo Cáo Thực Hành Cơ Sở Dữ Liệu**

**Đề Tài: Xây dựng hệ thống quản trị Dữ liệu Biển**

**Giảng Viên:** T.S Trần Việt Trung

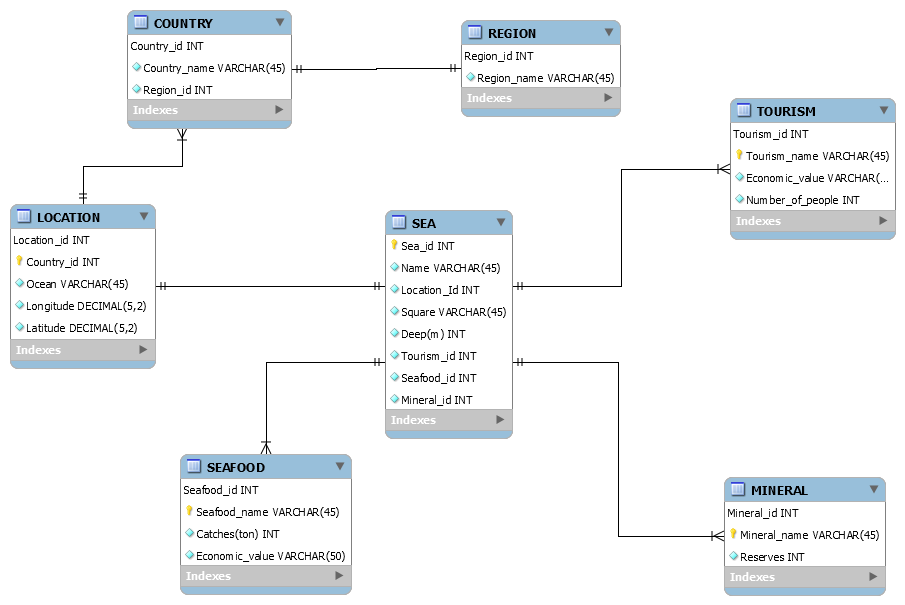
**Sinh viên:**Hoàng Đức Anh - 20176688   
Nguyễn Bảo Đức - 20176718  
Trần Minh Quang -20176858

1. **Giới thiệu bài toán**
2. **Ứng dụng của hệ CSDL**

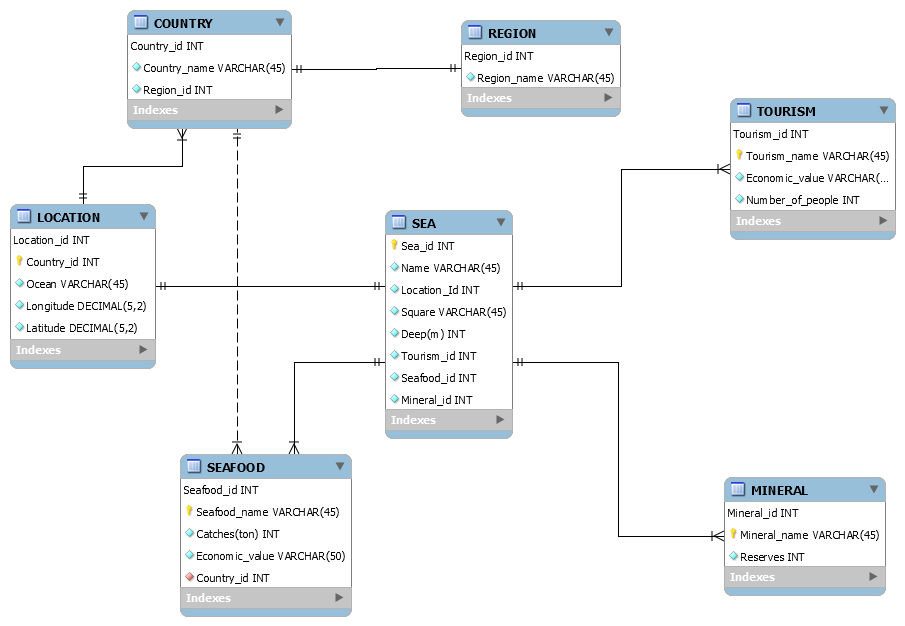
* Quản lý thông tin các biển trên thế giới

1. **Lược đồ quan hệ**

* **Lược đồ ban đầu**

****

* **Lược đồ sau khi Alter table**

****

**Chi tiết các bảng**

* **Bảng SEA:**
* **Khóa chính :** Sea\_id xác định các biển
* **Name** : Thuộc tính lưu tên của Biển
* **Square** : Thuộc tính lưu diện tích của biển
* **Deep** : Thuộc tính lưu độ sâu của biển
* **Bảng LOCATION:**
* **Khóa chính** : Location\_id và Country\_id xác định vị trí của Biển
* **Longitude** : Thuộc tính lưu kinh độ của vị trí
* **Latitude** : Thuộc tính lưu vĩ độ của vị trí
* **Ocean**  : Thuộc tính lưu xem biển thuộc đại dương nào
* **Khóa ngoại** : Location\_id tham chiếu tới khóa ngoại Location\_id của **bảng SEA**
* **Bảng COUNTRY:**
* **Khóa chính**  : Country\_id xác định Quốc gia
* **Country\_name** : Thuộc tính xác định tên của Biển
* **Region\_id** : Thuộc tính xác định khu vực của quốc gia
* **Khóa ngoại** : Country\_id tham chiếu tới khóa ngoại Country\_id của **bảng LOCATION**
* **Sau Khi Alter Table** : country\_id tham chiếu tới khóa ngoại Country\_id của bảng SEAFOOD
* **Bảng REGION:**
* **Khóa chính** : Region\_id xác định khu vực
* **Region\_name** : Xác định tên khu vực
* **Khóa ngoại** : Region\_id tham chiếu tới khóa ngoại Region\_id của **bảng COUNTRY**
* **Bảng TOURISM**
* **Khóa chính** : Tourism\_id và Tourism\_name xác định ngành du lịch
* **Economic\_value** : Thuộc tính xác định giá trị kinh tế của ngành du lịch
* **Number\_of\_people** : Thuộc tính xác định số lượng người của ngành du lịch
* **Khóa ngoại** : Tourism\_id tham chiếu tới khóa ngoại Tourism\_id của **bảng SEA**
* **Bảng SEAFOOD**
* **Khóa chính** : Seafood\_id và Seafood\_name xác định hải sản của biển
* **Economic\_value** : Thuộc tính xác định giá trị kinh tế của hải sản trên biển
* **Catches** : Thuộc tính xác định khối lượng đánh bắt hải sản của Biển
* **Khóa ngoại** : Seafood\_id tham chiếu tới khóa ngoại Seafood\_id của **bảng SEA**
* **Bảng MINERAL**
* **Khóa chính** : Mineral\_id và Mineral\_name xác định tài nguyên
* **Reserves**  : Thuộc tính xác định trữ lượng của tài nguyên
* **Khóa ngoại** : Mineral\_id tham chiếu tới khóa ngoại Mineral\_id **của bảng SEA**

1. **Các câu truy vấn**
2. **Thay đổi bảng SEAFOOD thêm 1 thuộc tính Country\_id**

alter table SEAFOOD

add

Country\_id int,

add constraint Exploited

foreign key (Country\_id) references COUNTRY(Country\_id);

1. **Update thêm id của Viet Nam vào bảng SEAFOOD ở 'The South China Sea' có sản lượng đánh bắt ‘Tuna’ lớn nhất**

update SEAFOOD

set

Country\_id = (select Country\_id from COUNTRY

where Country\_name = 'Vietnam')

where

Seafood\_id = (select Seafood\_id from SEA

where

sea\_name = 'The South China Sea')

and

Seafood\_name like '%Tuna%';

* **UPDATE**

1. **Cập nhật giá trị kinh tế du lịch của biển ‘Hudson Bay’ thành 500 nếu giá trị kinh tế du lịch của biển đó thấp hơn 500**

update

TOURISM

Set

Economic\_value = 500

where

Economic\_value<500

and

Tourism\_id = (select Tourism\_id from SEA where Sea\_name = "Hudson Bay");

1. **Cập nhật tên khoáng sản thành “Xang” của biển Hudson Bay nếu tên khoáng sản hiện tại không bắt đầu với “Na”**

update

MINERAL

Set

Mineral\_name = "Xang"

where

Mineral\_id = (select Mineral\_id from SEA where Sea\_name="Hudson Bay")

and not

Mineral\_name like "Na%";

1. **Cập nhật giá trị kinh tế (Economic\_value) của những thủy hải sản (SEAFOOD) có id lần lượt là 1, 6, 9 lên thêm lần lượt là 10%, 20%, 30%**

update

SEAFOOD

set

Economic\_value = case Seafood\_id

when 1 then Economic\_value + Economic\_value\*.1

when 6 then Economic\_value + Economic\_value\*.2

when 9 then Economic\_value + Economic\_value\*.3

else

Economic\_value

end;

* **SELECT**

1. **Đưa ra Đại dương có biển sâu nhất**

select

LOCATION.Ocean,SEA.sea\_name, SEA.Deep

from

LOCATION

join

SEA on SEA.Location\_id = LOCATION.Location\_id

order by Deep DESC

limit 1;

1. **Đưa ra tên biển đánh bắt nhiều “Grass Carp” nhất**

Select

SEA.sea\_name, SEAFOOD.Catches

From

SEA join SEAFOOD

on

SEA.Seafood\_id = SEAFOOD.Seafood\_id

where

SEAFOOD.Seafood\_name = ‘Grass Carp’;

1. **Đưa ra biển có dịch vụ tàu du lịch có doanh thu lớn nhất**

select

Sea\_name

from

SEA s

where

Tourism\_id = (select Tourism\_id from TOURISM t

where

Tourism\_name like '%Cruise Ships%'

order by

cast(t.Economic\_value as int) desc limit 1);

1. **Đưa ra tên châu lục có nhiều quốc gia gần biển nhất**

select

region\_name

from

REGION r

join COUNTRY c on (r.Region\_id = c.Region\_id)

group by

c.Region\_id

order by

count(\*) limit 1;

1. **Liệt kê các biển tiếp giáp Việt Nam**

select

sea\_name from SEA s

join

LOCATION l on (s.Location\_id = l.Location\_id)

join

COUNTRY c on (c.Country\_id = l.Country\_id)

where

Country\_name = 'Vietnam';

1. **Kể tên 2 biển ở Đại Tây Dương có trữ lượng khí tự nhiên lớn nhất**

select sea\_name

from

SEA s

join

LOCATION l on (s.Location\_id = l.Location\_id)

join

MINERAL m on (s.Mineral\_id = m.Mineral\_id)

where

l.Ocean like '%Atlantic%'

and

m.Mineral\_name like '%Natural gas%'

group by l.Location\_id

order by m.Reserves limit 2;

1. **Liệt kê các biển ở Thái Bình Dương mà không tiếp giáp Trung Quốc**

select

distinct sea\_name from SEA s

join

LOCATION l on (s.Location\_id = l.Location\_id)

where

l.Ocean like '%Pacific%'

and l.Location\_id

not in (select l.Location\_id from LOCATION l

join COUNTRY c on (l.Country\_id = c.Country\_id)

where Country\_name like '%China%');

1. **Liệt kê các biển nằm giữa chí tuyến Bắc và vòng Cực Bắc**

select

distinct sea\_name from SEA s

join

LOCATION l on (s.Location\_id = l.Location\_id)

where

l.Latitude between 23 and 66;

1. **Liệt kê các biển tiếp giáp nhiều hơn 5 nước**

select

s.sea\_name, count(l.Country\_id) as Countries

from

SEA s

join

LOCATION l on (s.Location\_id = l.Location\_id)

group by

l.Location\_id

having count(l.Location\_id) >= 5;

1. **Đưa ra ID, Tên, độ sâu, châu lục của các biển có độ sâu lớn hơn 1000m và có nước tiếp giáp nằm ở châu Á**

select

distinct S.Sea\_id, S.Sea\_name, S.Deep, R.Region\_name

from

SEA S

join LOCATION L on (S.Location\_id = L.Location\_id)

join COUNTRY C on (L.Country\_id = C.Country\_id)

join REGION R on (C.Region\_id = R.Region\_id)

where

R.Region\_name = "Asia" and S.Deep >1000

group by S.Sea\_id;

1. **Đưa ra ID, tên biển, Châu lục, Sản lượng đánh bắt thủy hải sản của biển có nước tiếp giáp nằm ở châu Á, mà có sản lượng thủy hải sản là lớn nhất**

select

distinct S.Sea\_id, S.Sea\_name, SF.Catches

from

SEA S

join SEAFOOD SF on (S.Seafood\_id = SF.Seafood\_id)

join LOCATION L on (S.Location\_id = L.Location\_id)

join COUNTRY C on (L.Country\_id = C.Country\_id)

join REGION R on (C.Region\_id = R.Region\_id)

where

R.Region\_name = "Asia"

having

SF.Catches = (select max(SF1.Catches) from SEAFOOD SF1) ;

1. **Đưa ra ID, Tên, Diện tích của biển có diện tích lớn thứ 3**

Select

distinct S1.Sea\_id, S1.Sea\_name, S1.Square

from

SEA S1

where 3 = (select count(distinct S2.Square)

from

SEA S2

where (S2.Square >= S1.Square))

order by

S1.Square desc;

1. **Đưa ra thông tin biển mà chữ cái thứ 2 không phải là chữ “e”**

select \* from SEA

where sea\_name not in (select Sea\_name from SEA where instr(Sea\_name, 'e') =2);

1. **Đưa ra ID, Tên biển, Các nước tiếp giáp, và tổng sản lượn của các biển.**

select

distinct S.Sea\_id, S.Sea\_name, group\_concat(distinct C.Country\_name, " ") as Country, sum(distinct Reserves)

from SEA S

join

MINERAL M on (S.Mineral\_id = M.Mineral\_id)

join

LOCATION L on (S.Location\_id = L.Location\_id)

join

COUNTRY C on ( L.Country\_id = C.Country\_id)

group by S.Sea\_id;

1. **Liệt kê các biển và trữ lượng dầu mỏ (Nhiều- Trung bình – ít)**

select s.sea\_name,

(case when m.Reserves > 300000 then 'Dau mo nhieu' when m.Reserves > 100000 then 'Dau mo trung binh' when m.Reserves > 000000 then 'Dau mo it' end ) as Resources

from

SEA s

join

MINERAL m on (s.Mineral\_id = m.Mineral\_id)

where

m.Mineral\_name like '%Petroleum%';

1. **Lấy ra tên biển tiếp giáp nhiều quốc gia nhất**

Select s.sea\_name, count(l.Country\_id) as Countries

from SEA s

join

LOCATION l on (s.Location\_id = l.Location\_id)

group by l.Location\_id

having count(l.Location\_id) >= all

(select count(Location\_id) from LOCATION group by Location\_id);

1. **Liệt ra tên các nước tiếp giáp với ‘The South China Sea’**

select

Country\_name from COUNTRY c

join

LOCATION l on (c.Country\_id = l.Country\_id)

join

SEA s on (l.Location\_id = s.Location\_id)

where

s.sea\_name like '%The South China Sea%';

1. **Lập ma trận tọa độ tương đối của các biển**

select s.sea\_name,l.latitude,

(case when l.Longitude > -180 and l.Longitude < -90 then '\*' else ' 'end) as '90-',

(case when l.Longitude > -90 and l.Longitude < 00 then '\*' else ' ' end) as '0-',

(case when l.Longitude > 00 and l.Longitude < 90 then '\*' else ' ' end) as '0+',

(case when l.Longitude > 90 then '\*' else ' 'end) as '90+'

from SEA s

join

LOCATION l on (s.Location\_id = l.Location\_id)

group by l.Location\_id

order by l.Latitude desc;

1. **Đưa ra tổng giá trị kinh tế ngành du lịch của các biển thuộc "Pacific Ocean"**

Select

distinct S.Sea\_id, S.Sea\_name, sum(distinct T.Economic\_value)

from

SEA S join TOURISM T on (S.Tourism\_id = T.Tourism\_id)

join

LOCATION L on (S.Location\_id=L.Location\_id)

where L.Location\_id

in

(select L1.Location\_id from LOCATION L1 where L1.Ocean like "%Pacific Ocean%")

group by S.Sea\_id ;

1. **Đưa ra biển tiếp giáp nhiều hơn 5 quốc gia và có sản lượng đánh bắt hải sản là lớn nhất**

select

s.sea\_name, sum(distinct Catches) as Total

from SEA s

join

LOCATION l on (s.Location\_id = l.Location\_id)

join

SEAFOOD sf on (s.Seafood\_id = sf.Seafood\_id)

group by l.Location\_id

having count(l.Location\_id) >= 5

order by sum(distinct Catches) desc limit 1;

1. **Biển "Mediterranean Sea" tiếp giáp với nhiều quốc gia ở khu vực nào nhất**

select Region\_name from REGION r

join COUNTRY c

on (r.Region\_id = c.Region\_id)

join LOCATION l

on (c.Country\_id = l.Country\_id)

where l.Location\_id = (select Location\_id from SEA

where sea\_name like '%Mediterranean%')

group by c.Region\_id

order by count(distinct l.Country\_id) desc limit 1;

1. **Đưa ra tổng “Natural gas” của các biển trong “Atlantic Ocean”**

select L.Ocean, sum(distinct M.Reserves) as "Atlantic Ocean's Reserves of Natural Gas" from SEA S join MINERAL M on (S.Mineral\_id = M.Mineral\_id) join LOCATION L on (S.Location\_id = L.Location\_id) where M.Mineral\_name = "Natural gas" and L.Ocean = "Atlantic Ocean";

1. Khu vực Europe có những biển nào

select S.Sea\_id, S.Sea\_name, R.Region\_name from SEA S join LOCATION L on (S.Location\_id = L.Location\_id) join COUNTRY C on (C.Country\_id = L.Country\_id) join REGION R on ( R.Region\_id = C.Region\_id) where R.Region\_id = (select R1.Region\_id from REGION R1 where R1.Region\_name like "%Europe%");

1. Đưa ra tổng giá trị kinh tế hải sản và du lịch của tất cả các

Select distinct S.Sea\_id, S.Sea\_name, sum(distinct T.Economic\_value) as "Tourism\_value", sum(distinct SF.Economic\_value) as "Seafood\_value", sum(distinct T.Economic\_value) + sum(distinct SF.Economic\_value) as "Total of Economic value" from SEA S join TOURISM T on (S.Tourism\_id = T.Tourism\_id) join SEAFOOD SF on (S.Seafood\_id = SF.Seafood\_id) group by S.Sea\_id;

1. **Xóa bản ghi của bảng SEA mà có tiếp giáp với Trung Quốc và có trữ lượng khí đốt là lớn nhất.**

delete from SEA

where sea\_name =

(select sea\_name from (select \* from SEA) s

join LOCATION l on (s.Location\_id = l.Location\_id)

join MINERAL m on (s.Mineral\_id = m.Mineral\_id)

join COUNTRY c on (l.Country\_id = c.Country\_id)

where c.Country\_name like '%China%'

and m.Mineral\_name like '%Natural gas%'

order by m.Reserves desc limit 1);