**BỘ GIÁO DỤC VÀ ĐÀO TẠO**Shape, square

Description automatically generated

**TRƯỜNG ĐẠI HỌC XÂY DỰNG**

----------



**BÁO CÁO**

**Đồ Án Cơ Sở Dữ Liệu**

***Đề tài: Quản lý ký túc xá.***

**Lớp : 64PM2**

**Nhóm 4 :**

Đoàn Văn Hiệp 1517264

**Giảng viên hướng dẫn :** GV Nguyễn Đình Anh.

***Hà Nội, 11/2022***

1. Giới thiệu chung.
2. Nêu vấn đề.

Hàng năm mỗi trường đại học, cao đẳng tiếp nhận hàng nghìn học sinh, sinh viên. Cùng với đó nhu cầu về việc đăng kí nội trú trong nhà trường ngày càng cao. Trước bài toán đặt ra với các trường đại học, cao đẳng hiện nay: vấn đề quản lý kí túc xá – một vấn đề đã có từ lâu nhưng vẫn còn khá nhiều bất cập như: việc quản lý ở nhiều trường vẫn còn theo phương pháp thủ công, các dữ liệu không có tính thống nhất, chặt chẽ. Trước thực trạng nhiều trường đại học hiện nay với lượng sinh viên trọ ở kí túc xá rất đông nhưng vẫn có hình thức quản lý chính là thực hiện thủ công trên giấy tờ trong khi chỉ có 1 đến 2 nhân viên quản lý kí túc xá khiến cho khối lượng công việc của họ thực sự nhiều lúc quá lớn và hiệu quả không được cao.Hiện nay không còn mấy ai xa lạ với những sản phẩm và ứng dụng của công nghệ thông tin – ngành khoa học đang trở thành một phần tất yếu trong cuộc sống hiện nay. Cùng với sự phát triển không ngừng đó, những ứng dụng của công nghệ thông tin trong việc quản lý nhân sự, quản lý nhà hàng, quản lý trường học … đã và đang góp phần giảm thiểu được lượng chi phí đầu tư cho nguồn nhân công, đồng thời tăng tính hiệu quả chính xác trong việc khai thác và quản lý dữ liệu.Từ thực tế đó đã đặt ra câu hỏi: Tại sao lại không áp dụng công nghệ thông tin vào quản lý kí túc xá? Từ câu hỏi đó nhóm chúng tôi đã bắt tay vào tìm hiểu và nghiên cứu đề tài : “Xây dựng phần mềm quản lý kí túc xá” với hi vọng sẽ là cơ sở phát triển sau này cho việc quản lý kí túc xá nói chung và cho từng trường nói riêng. Mục tiêu và phạm vi đề tài.

1. Mục tiêu.

Xây dựng một phần mềm quản lý kí túc xá bước đầu giúp cho người quản lý có thể tiếp cận dễ dàng ứng dụng Công nghệ thông tin để áp dụng vào quá trình quản lý để mang lại hiệu quả cao hơn đồng thời giảm tải được khối lượng công việc.

Mục tiêu xây dựng phần mềm:

* Quản lý lượng sinh viên nội trú bao gồm việc tiếp nhận sinh viên nội trú, thêm mới, xóa , sửa thông tin sinh viên nội trú.
* Quản lý các tòa nhà bao gồm việc quản lý các phòng thuộc tòa nhà đó. Quản lý lượng sinh viên của từng phòng, các trang thiết bị,…
* Quản lý hợp đồng khi có sinh viên có nguyện vọng muốn vào ký túc xá. Trong thời gian hợp đồng, nếu sinh viên vi phạm quy định của ký túc xá và đủ cơ sở để buộc ra khỏi ký túc xá thì có thể hủy hợp đồng.
* Quản lý thu chi, bao gồm phiếu thu phí điện nước, phiếu phạt đối những phòng hoặc sinh viên làm mất mát hư hại cơ sở vất chất của ký túc xá, phiếu chi những khoản như điện nước, trả lương cho nhân viên.

1. Các vấn đề cần xử lý trong bài toán “Quản lý ký túc xá sinh viên”.
2. Quản lý sinh viên.

* Tạo mới sinh viên nội trú
* Sửa thông tin sinh viên
* Xóa sinh viên (trong trường hợp sinh viên hết hợp đồng và không có nhu cầu ở lại hoặc sinh viên vi phạm quy định buộc rời ký túc xá).

1. Quản lý tòa nhà / phòng.

* Tạo mới tòa nhà/phòng (trong trường hợp xây mới tòa nhà và phát sinh thêm cái phòng mới kèm theo).
* Sửa thông tin các phòng (số lượng sinh viên tối đa, giá phòng, mô tả các trang thiết bị).
* Xóa tòa nhà/phòng trong trường hợp tháo dỡ hoặc không còn thuộc quyền quản lý của ký túc xá.

1. Quản lý hợp đồng.

* Tạo mới hợp đồng ( dành cho các sinh viên muốn vào nội trú hoặc các sinh viên muốn gia hạn ).
* Hủy hợp đồng (trong trường hợp sinh viên bị buộc rời ký túc xá).

1. Quản lý thu chi.

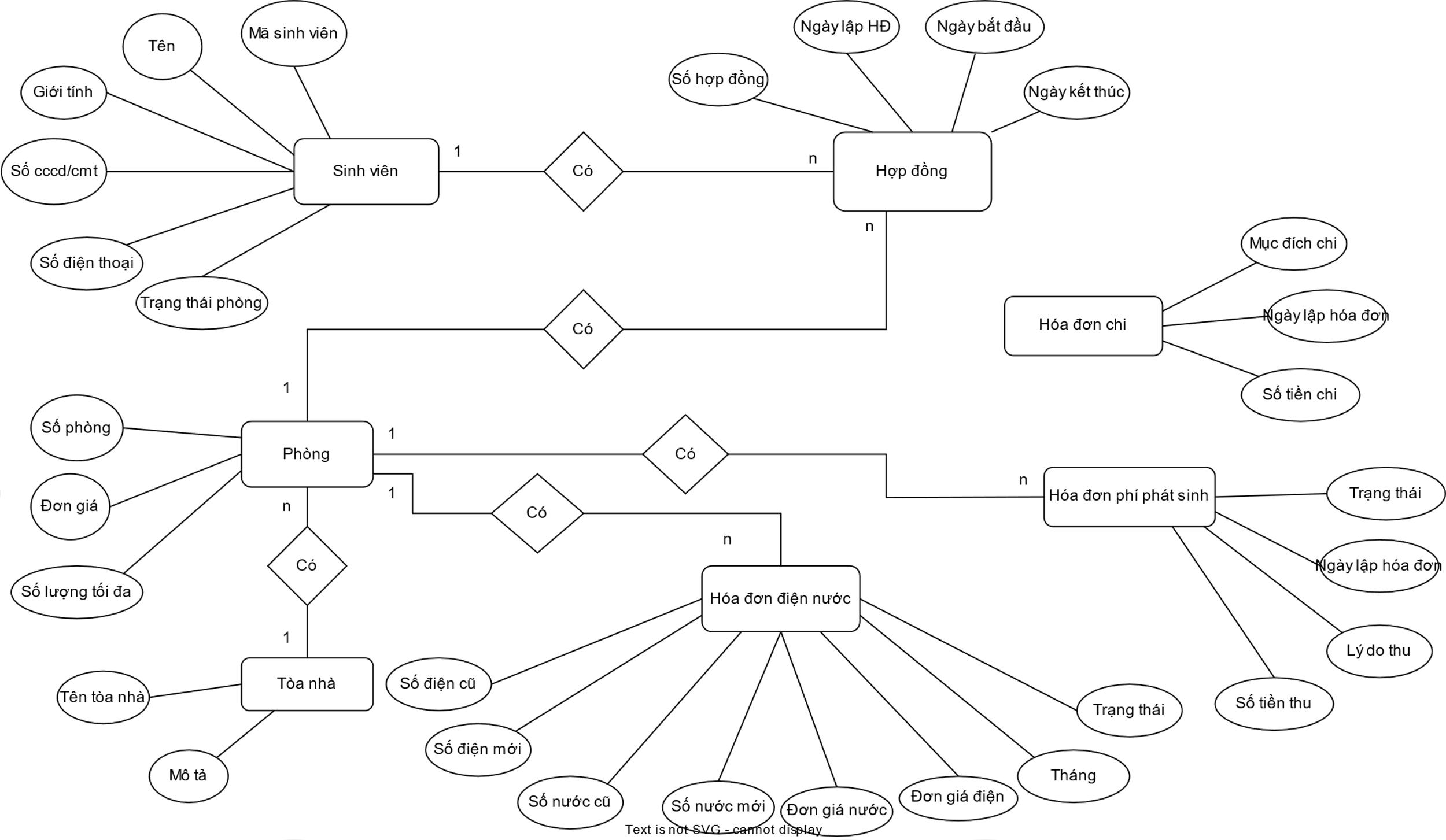
* Tạo mới các phiếu thu/chi.
* Cập nhật trạng thái phiếu thu.

1. Báo cáo thống kê.

* Báo cáo tổng thu chi theo thời gian tìm kiếm.
* Báo cáo tình trạng các phòng.

1. Thiết kế cơ sở dữ liệu.

1, Mô hình thực thể liên kết ERD:



2, Tổng hợp các thuộc tính và khóa chính:

* Tòa nhà (Id , tên tòa nhà, mô tả)
* Phòng ( Id, mã tòa nhà, số phòng, giá phòng, số lượng tối đa)
* Sinh viên ( Id, mã sinh viên, tên, giới tính, số cccd, số điện thoại)
* Hợp đồng ( Id, số hợp đồng, mã sinh viên, mã phòng, ngày lập hđ, ngày bắt đầu, ngày kết thúc)
* Hóa đơn điện nước ( Id, mã phòng, số điện cũ, số điện mới, số nước cũ, số nước mới, đơn giá điện, đơn giá nước, tháng hóa đơn, trạng thái)
* Hóa đơn phí phát sinh ( Id, mã phòng, lý do thu, số tiền thu, ngày lập hóa đơn, trạng thái)
* Hóa đơn chi ( Id, mục địch chi, số tiền chi, ngày lập hđ)

3, Các ràng buộc:

* Trong thuộc tính của hợp đồng, ngày bắt đầu luôn luôn phải nhỏ hơn ngày hết thúc.
* Trong thuộc tính của hóa đơn điện nước, số điện đầu luôn phải nhỏ hơn số điện cuối và số nước đầu luôn phải nhỏ hơn số nước cuối.
* Khi tạo mới hợp đồng, ngày tạo hợp đồng không được nhỏ hơn ngày hiện hành.
* Khi tạo mới hóa đơn phí phát sinh và hóa đơn chi, ngày lập hóa đơn không được nhỏ hơn ngày hiện hành.

4, Các rang buộc khóa ngoại:

* Trong Phòng có mã tòa nhà làm khóa ngoại
* Trong Hợp Đồng có mã sinh viên, mã phòng làm khóa ngoại
* Trong Hóa đơn điện nước có mã phòng làm khóa ngoại
* Trong Hóa đơn chi có mã phòng làm khóa ngoại

5, Xây dựng sơ sở dữ liệu :

5.1, Bảng tòa nhà (Building):

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Column Name | Data Type | Description |
| 1 | Id | Int NOT NULL primary key | Mã tòa nhà |
| 2 | Name | nvarchar(64) NOT NULL | Tên tòa nhà |
| 3 | Description | nvarchar(255) NULL | Mô tả |

5.2, Bảng Phòng (Room):

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Column Name | Data Type | Description |
| 1 | Id | Int NOT NULL primary key | Mã phòng |
| 2 | BuildingId | int NOT NULL Foreign key | Mã tòa nhà |
| 3 | RoomNo | nvarchar(64) NOT NULL | Tên phòng |
| 4 | UnitPrice | bigint NOT NULL | Giá phòng |
| 5 | Maxx | int NOT NULL | Số lượng tối đa |
| 6 | Amount | int NOT NULL  Check ( Amount < Maxx) | Số lượng hiện tại |

5.3, Bảng hóa đơn điện nước (Invoice):

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Column Name | Data Type | Description |
| 1 | Id | Int NOT NULL primary key | Mã hóa đơn |
| 2 | RoomId | int NOT NULL Foreign key | Mã phòng |
| 3 | ElecOld | int NOT NULL | Số điện cũ |
| 4 | ElecNew | int NOT NULL  Check(ElecNew>ElecOld) | Số điện mới |
| 5 | WarOld | int NOT NULL | Số nước cũ |
| 6 | WarNew | int NOT NULL Check (WarNew >ElecOld) | Số nước mới |
| 7 | ElecUnitPrice | int NOT NULL | Đơn giá điện |
| 8 | WarUnitPrice | int NOT NULL | Đơn giá nước |
| 8 | Month | datetime2(7) NOT NULL | Tháng hóa đơn |
| 9 | IsPaid | bit NOT NULL | Trạng thái thanh toán |

5.4, Bảng hóa đơn phí phát sinh (InBill):

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Column Name | Data Type | Description |
| 1 | Id | Int NOT NULL primary key | Mã hóa đơn |
| 2 | RoomId | int NOT NULL Foreign key | Mã phòng |
| 4 | Cost | bigint NOT NULL | Số tiền thu |
| 5 | Date | datetime2(7) NOT NULL | Ngày lập hđ |
| 6 | Reason | nvarchar(500) NULL | Lý do lập hđ |
| 7 | IsPaid | bit NOT NULL | Trạng thái thanh toán |

5.5, Bảng hóa đơn chi (OutBill):

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Column Name | Data Type | Description |
| 1 | Id | Int NOT NULL primary key | Mã hóa đơn |
| 2 | Cost | bigint NOT NULL | Số tiền chi |
| 3 | Date | datetime2(7) NOT NULL | Ngày lập |
| 4 | Description | nvarchar(500) NULL | Ghi chú chi tiêu |

5.5, Bảng sinh viên (Student):

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Column Name | Data Type | Description |
| 1 | Id | Int Not Null primary key | Mã sinh viên |
| 2 | Name | nvarchar(255) Not Null | Tên sinh viên |
| 3 | StudentNo | Int Not Null | Mã số sinh viên |
| 4 | Gender | Int Not Null | Giới tính |
| 4 | Cmnd | nvarchar(64) NOT NULL | Số căn cước |
| 5 | Phone | nvarchar(64) Null | Số điện thoại |
| 6 | IsRoom | Bit Not Null | Trạng thái có phòng |

5.6, Bảng hợp đồng (Contract):

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Column Name | Data Type | Description |
| 1 | Id | Int NOT NULL primary key | Id |
| 2 | ContractNo | nvarchar(255) NOT NULL | Mã hợp đồng |
| 3 | StudentId | int NOT NULL Foreign key | Mã sinh viên |
| 4 | RoomId | int NOT NULL Foreign key | Mã phòng |
| 5 | ContractDate | datetime2(7) NOT NULL | Ngày lập HĐ |
| 6 | StartDate | datetime2(7) NOT NULL | HĐ từ ngày |
| 7 | EndDate | datetime2(7) NOT NULL  Check (EndDate>StartDate) | Đến ngày |

1. Mô tả ứng dụng và các truy vấn đã sử dụng.

|  |  |  |
| --- | --- | --- |
| Vấn đề | Giải pháp | Truy vấn sử dụng |
| Xem danh sách sinh viên hoặc tìm kiếm | Sử dụng Stored Procedure | Create procedure GetAllOrFilterStudent  @Name nvarchar(255),  @Gender int,  @StudentNo nvarchar(64),  @Cmnd nvarchar(64),  @Phone nvarchar(64),  @IsRoom bit As  Select Id, Name, StudentNo, Gender, Cmnd, Phone, IsRoom From Student  Where IsDeleted=0 and  (@Name IS null or Name like '%'+@Name+'%') and  (@StudentNo is null Or StudentNo like '%'+@StudentNo+'%') and  (@Gender is null or Gender = @Gender) and  (@Cmnd is null or Cmnd like '%'+@Cmnd+'%') and  (@Phone is null or Phone like '%'+@Phone+'%') and  (@IsRoom is null or IsRoom = @IsRoom) |
| Xem danh sách sinh viên trong 1 phòng bất kì | Sử dụng Stored Procedure | Create procedure GetListStudentInRoom  @RoomId int  AS  Begin  Select Student.Id, Student.Name as Name, Gender, StudentNo, Cmnd, Phone from Student INNER JOIN Contract on Student.Id =Contract.StudentId  where RoomId = @RoomId and Student.IsDeleted=0 and Contract.IsDeleted=0 and Contract.EndDate > = GETDATE()  End |
| Xem danh sách hoặc tìm kiếm hợp đồng | Sử dụng Stored Procedure | Create Procedure GetAllOrFilterContract  @RoomId int,  @StartDate nvarchar(50),  @EndDate nvarchar(50)  As Begin  Select Contract.Id, ContractNo, StudentNo, Student. Name As StudentName, Gender, Phone, RoomNo, UnitPrice, ContractDate, StartDate, EndDate  from Contract  inner join Student on Student.Id = Contract.StudentId  inner join Room on Room.Id = Contract.RoomId  where Contract.IsDeleted = 0 and  (@RoomId Is null or RoomId = @RoomId) and (Contract.ContractDate between @StartDate and @EndDate )  End |
| Lấy danh sách các phòng còn chỗ trống và phù hợp với tính của sinh viên muốn vào ở | Sử dụng Stored Procedure | Create Procedure GetListRoomForConTract  @StudentId int  As Begin  Declare @Gender int  Set @Gender = ( Select Gender From Student where Id = @StudentId)  Select Room.Id, RoomNo, UnitPrice, Maxx-Amount AS Empty from Room  Left Join Contract on Room.Id = Contract.RoomId  Left join Student on Student.Id = Contract.StudentId  where Amount < Maxx and Room.Id Not In  (Select RoomId from Contract Inner Join Student on Contract.StudentId = Student.Id where Gender <>@Gender)  order by Empty  End |
| Lấy danh sách các phòng để tạo hóa đơn điện nước (chỉ lấy những phòng chưa hóa đơn ) | Sử dụng Store Procedure | Create Procedure GetListRoomForCreateInvoice  As  Begin  Select Id,RoomNo from Room where IsDeleted = 0 and Id Not In  (Select Room.Id From Room inner join Invoice on Room.Id = Invoice.RoomId  Where MONTH(Invoice.Month) = MONTH(GetDate()) And YEAR( Invoice.Month ) = YEAR(GetDate()))  End |
| Xem danh sách hoặc tìm kiếm hóa đơn điện nước | Sử dụng Store Procedure | Create Procedure GetAllOrFilterInvoice  @RoomId int,  @StartDate nvarchar(50),  @EndDate nvarchar(50),  @IsPaid bit  As Begin  Select Invoice.Id AS Id,RoomId, Room.RoomNo As RoomNo, ElecOld, ElecNew, WarOld, WarNew, ElecUnit, WarUnit, Month, IsPaid  From Invoice Inner Join Room on Room.Id = Invoice.RoomId where Invoice.IsDeleted = 0 and  (@RoomId IS null or RoomId = @RoomId) and  (Invoice.Month between @StartDate and @EndDate )  and (@IsPaid is null or IsPaid = @IsPaid)  End |
| Lấy danh sách các phòng để tạo hóa đơn điện nước (chỉ lấy những phòng chưa hóa đơn ) | Sử dụng Store Procedure | Create Procedure GetListRoomForCreateInvoice  As  Begin  Select Id,RoomNo from Room where IsDeleted = 0 and Id Not In  (Select Room.Id From Room inner join Invoice on Room.Id = Invoice.RoomId  Where MONTH(Invoice.Month) = MONTH(GetDate()) And YEAR( Invoice.Month ) = YEAR(GetDate()))  End |
| Xem danh sách hoặc tìm kiếm phiếu thu | Sử dụng query | Create Procedure GetAllOrFilterInBill  @RoomId int,  @StartDate nvarchar(50),  @EndDate nvarchar(50),  @IsPaid bit  As  Begin  Select InBill.Id, RoomNo, Cost, Date, Reason, IsPaid from InBill  Inner Join Room on Room.Id = InBill.RoomId  Where InBill.IsDeleted =0 and Room.IsDeleted=0 and  (@RoomId Is null or RoomId = @RoomId) and  (InBill.Date between @StartDate and @EndDate ) and  (@IsPaid is null or IsPaid = @IsPaid)  End |
| Xem danh sách hoặc tìm kiếm phiếu chi | Sử dụng Stored Procedure | Create Procedure GetAllOrFilterOutBill  @StartDate nvarchar(50),  @EndDate nvarchar(50)  As  Begin  Select Id,Cost,Date,Description from OutBill where IsDeleted =0 and (Date between @StartDate and @EndDate)  End |
| Xem tổng số hợp đồng theo tháng | Sử dụng Stored Procedure | Create Procedure GetContractForDashboard  @Date nvarchar(50)  As  Begin  SELECT  DatePart (MONTH, ContractDate) AS Month,  COUNT(\*) AS Total  FROM Contract  where YEAR(ContractDate)=YEAR(@Date) and IsDeleted=0  GROUP BY DatePart (MONTH, ContractDate)  ORDER BY DatePart (MONTH, ContractDate)  End |
| Tổng chi theo tháng | Sử dụng Stored Procedure | Create Procedure GetOutBillForDashboard  @Date nvarchar(50)  As  Begin  SELECT  DatePart (MONTH, Date) AS Month,  SUM(OutBill.Cost ) AS Total  FROM OutBill where IsDeleted =0 and YEAR(Date)=YEAR(@Date)  GROUP BY DatePart (MONTH, Date)  ORDER BY Month  End |
| Tổng thu điện nước theo tháng | Sử dụng Stored Procedure | Create Procedure GetInvoiceForDashboard  @Date nvarchar(50)  As  Begin  SELECT  DatePart (MONTH, Month) AS Month,  SUM((ElecNew - ElecOld) \* ElecUnit + (WarNew - WarOld) \* WarUnit) AS Total  FROM Invoice where IsDeleted =0 and YEAR(Month)=YEAR(@Date)  GROUP BY DatePart (MONTH, Month)  ORDER BY Month  End |
| Tổng thu phí phát sinh theo tháng | Sử dụng Stored Procedure | Create Procedure GetInBillForDashboard  @Date nvarchar(50)  As  Begin  SELECT  DatePart (MONTH, Date) AS Month,  SUM(InBill.Cost ) AS Total  FROM InBill where IsDeleted =0 and YEAR(Date)=YEAR(@Date)  GROUP BY DatePart (MONTH, Date)  ORDER BY Month  End |
| Tổng thu hợp đồng theo tháng | Sử dụng Stored Procedure | Create Procedure GetTotalMoneyContract  @Date nvarchar(50)  As  Select  DatePart(MONTH, ContractDate) AS Month,  Sum(Room.UnitPrice) AS TotalMoney  From Contract INNER JOIN Room ON Contract.RoomId=Room.Id  where YEAR(ContractDate)=YEAR(@Date) and Contract.IsDeleted=0  GROUP BY DatePart (MONTH, ContractDate)  ORDER BY DatePart (MONTH, ContractDate) |
| Xét hợp đồng hết hạn đưa trạng thái sinh viên về chưa có phòng và tính lại số sinh viên trong phòng hợp đồng | Sử dụng Stored Procedure | Create Procedure contract\_timeout  AS  begin  Declare @RoomId int  Declare cursorContract  Cursor for  Select RoomId from Contract  where EndDate < GetDate()  AND IsDeleted =0  Open cursorContract  Fetch next FROM  cursorContract -- Đọc dòng đầu tiên  into @RoomId while @@FETCH\_STATUS = 0  begin  update Room  SET Amount = (  Select Count(Id) from Contract  where IsDeleted =0 and RoomId = @RoomId and Contractedness >= GetDate())  where Id = @RoomId  Fetch next from cursorContract  into @RoomId  end  Close cursorContract -- Đóng Cursor  DealLocate cursorContract  Update Student set IsRoom = 0  where Id in (Select StudentId from Contract  where EndDate < GetDate() and IsDeleted =0) and IsDeleted =0) and Id not in  (select StudentId from Contract WHERE EndDate > GETDATE() AND IsDeleted =0)  end |
| Cập nhật thông tin phòng ở ( số lượng sinh viên, số giường trống) và thông tin sinh viên ( đã có phòng ) sau khi tạo hợp đồng | Sử dụng Strigger | Create trigger trg\_contract On CONTRACT after insert  as begin  Declare @StudentId int = (Select StudentId from inserted)  Declare @RoomId int = (Select RoomId from inserted)  Update Room Set Amount = ( Select Count(Id) from Contract where IsDeleted =0 and RoomId = @RoomId And EndDate >= GETDATE()) where Id = @RoomId  Update Student Set IsRoom = 1 where Student.Id = @StudentId  End |
| Cập nhật số lượng sinh viên trong 1 phòng, tình trạng của sinh viên đó | Sử dụng Trigger | Create Trigger trg\_contract\_delete On Contract After Update  As Begin  Declare @dele bit = (Select IsDeleted from inserted)  Declare @StudentId int = (Select StudentId from deleted)  Declare @RoomId int = (Select RoomId from deleted)  Update room set Amount = (Select Count(Id) from Contract where IsDeleted =0 and RoomId = @RoomId) where Id = @RoomId  If(@dele = 1)  Begin  Update Student Set IsRoom = 0  where Student.Id = @StudentId  End  Else  Begin  Update Student Set IsRoom = 1  where Student.Id = @StudentId  End  End |
| Thêm mới sinh viên | Sử dụng query | Insert into Student (IsDeleted, Name, Gender, StudentNo, Cmnd, Phone, IsRoom) Values (0, N’?’, ? , ? , ? , ?, 0) |
| Sửa thông tin sinh viên | Sử dụng query | Update Student Set Name = N’?’, StudentNo = ?, Gender = ?, Cmnd = ?, Phone = ?, IsRoom = ? Where Id = ? |
| Xóa sinh viên ( xóa mềm ) | Sử dụng query | Update Student Set IsDeleted = 1, DeletionTime = GetDate() Where Id =? |
| Xem danh sách các tòa nhà | Sử dụng query | Select Id ,Name, Description From Building where IsDeleted = 0 |
| Thêm mới tòa nhà | Sử dụng query | Insert into Building (Name, Description, IsDeleted)  Values ( N’ ? ‘, N ‘?’, 0) |
| Sửa thông tin tòa nhà | Sử dụng query | Update Building Set Name = N' ? ', Description = N' ? ‘Where Id = ? |
| Xóa tòa nhà | Sử dụng query | Update Building Set IsDeleted = 1, DeletionTime = GetDate() where Id = ?  Update Room Set IsDeleted = 1, DeletionTime = GetDate() where BuildingId = ? |
| Xem danh sách phòng ở theo tòa nhà | Sử dụng query | Select Id ,RoomNo, UnitPrice, Maxx, Amount From Room where IsDeleted = 0 And BuildingId = ? |
| Thêm phòng mới | Sử dụng query | Insert into Room (IsDeleted, BuildingId, RoomNo, UnitPrice, Maxx, Amount) Values (0, ? , N’ ? ‘, ? , ? , 0 ) |
| Sửa thông tin phòng | Sử dụng query | Update Room Set RoomNo = N’ ? ‘, UnitPrice = ?, Maxx = ? Where Id = ? |
| Xóa phòng | Sử dụng query | Update Room Set IsDeleted = 1, DeletionTime =GetDate() |
| Tạo hợp đồng mới | Sử dụng query | Insert into Contract ( IsDeleted, ContractNo, StudentId, RoomId, ContractDate, StartDate, EndDate )  Values ( 0, ? , ? , ? , GetDate(), ? , ? ) |
| Sửa thông tin hợp đồng | Sử dụng query | Update Contract Set RoomId = ? , StartDate = ?, EndDate = ? where Id = ? |
| Xóa hợp đồng | Sử dụng query | Update Contract Set IsDeleted = 1, DeletionTime = GetDate() where Id = ? |
| Thêm mới hóa đơn điện nước | Sử dụng query | Insert into Invoice ( IsDeleted, RoomId, ElecOld, ElecNew, WarOld, WarNew, ElecUnit, WarUnit, Month, IsPaid )  Values (0 , ? , ? , ? , ? , ? , ?, ?, GetDate(), 0 ) |
| Sửa thông tin hóa đơn điện nước | Sử dụng query | Update Invoice Set ElecOld = ?, ElecNew = ?, WarOld = ?, WarNew =? , ElecUnit = ?, WarUnit = ? , IsPaid = ? Where Id = ? |
| Xóa hóa đơn điện nước | Sử dụng query | Update Invoice Set IsDeleted = 1, DeletionTime = GetDate() Where Id =? |
| Thêm mới phiếu thu | Sử dụng query | Insert into InBill (IsDeleted, RoomId, Cost, Date, Reason, IsPaid) Values (0, ?, ?, GetDate(), N’?’, 0) |
| Sửa phiếu thu | Sử dụng query | Update InBill Set Cost = ?, Reason = ?, IsPaid = ? where Id = ? |
| Xóa phiếu thu | Sử dụng query | Update InBill Set IsDeleted = 1, DeletionTime = GetDate() where Id = ? |
| Thêm mới phiếu chi | Sử dụng query | Insert into OutBill (IsDeleted, Cost, Date, Description)  Values ( 0, ? ,GetDate(), N’?’) |
| Sửa phiếu chi | Sử dụng query | Update OutBill Set Cost = ?, Description = N’?’ where Id = ? |
| Xóa phiếu chi | Sử dụng query | Update OutBill Set IsDeleted = 1, DeletionTime = GetDate() where Id = ? |