Пагинация

* **Skip** – для пропуска строк
* **Take** – вернут указанное количество строк

// пагинация

Console.WriteLine("Пагинация");

int pageSize = 3;

int pageIndex = 1;

games = games.Skip(pageSize \* (pageIndex - 1)).Take(pageSize);

// проекция(select) - используется для оптимизации, позволяет вернуть только требуемые данные

var gamesData = games

.Select(g => new

{

GameName = g.Name

});

Console.WriteLine(gamesData.ToQueryString());

**Для проекции можно создать свой тип данных**

**Пример:**

* **Join** – позволяет объединить несвязанные таблицы
* **GroupBy** – позволяет группировать данные и работать с агрегатными функциями
* По умолчанию данные кэшируются и отслеживаются
* Способы отключения отслеживания изменений

var games = context.Games.AsNoTracking();

**Методы для изменения объектов не требующих загрузки данных**

context.НаборДанных

.Where(условие отбора изменяемых записей) // опционально

.ExecuteОперация(…);

context.Categories.Where(c => c.CategoryId > 10).ExecuteDelete();