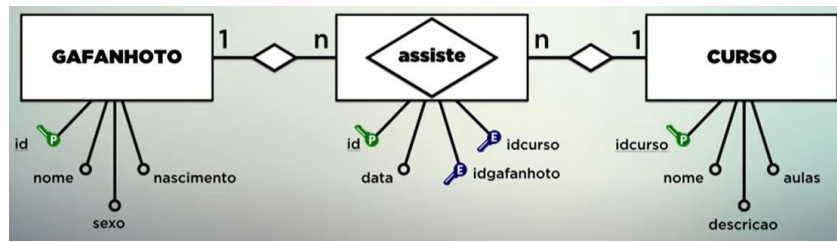


AULA 16 – INNER JOIN COM VÁRIAS TABELAS

Exemplo MUITOS-PARA-MUITOS



Criando a tabela extra

O nome dessa tabela pode ser qualquer

```
CREATE TABLE gafanhoto_assiste_curso (  
  id int NOT NULL AUTO_INCREMENT,  
  data date,  
  idgafanhoto int,  
  idcurso int,  
  PRIMARY KEY (id),  
  FOREIGN KEY (idgafanhoto) REFERENCES gafanhotos(id),  
  FOREIGN KEY (idcurso) REFERENCES cursos(idcurso)  
) DEFAULT CHARSET = utf8;
```

Aqui eu coloco os

Aqui eu defino minha chave primária

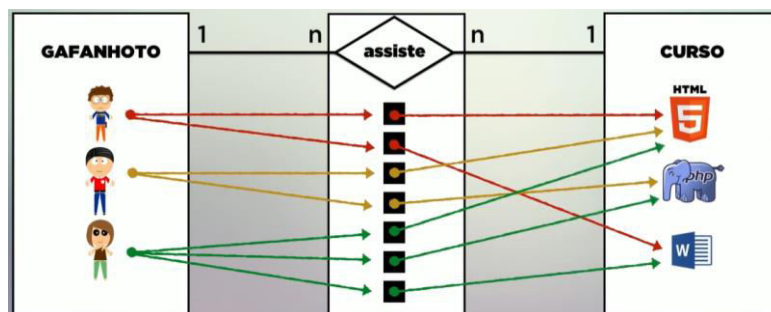
Aqui eu defino minhas chaves estrangeiras

No MySQL:

```
CREATE TABLE gafanhotos_assiste_curso (  
  id INT NOT NULL AUTO_INCREMENT,  
  data DATE,  
  idgafanhoto INT,  
  idcurso INT,  
  PRIMARY KEY (id),  
  FOREIGN KEY (idgafanhoto)  
    REFERENCES gafanhotos (id),  
  FOREIGN KEY (idcurso)  
    REFERENCES cursos (idcurso)  
) DEFAULT CHARSET=UTF8;
```

→ Agora com a tabela pronta, como nós colocamos o dado lá dentro? Que tipo de relações iremos fazer?

Entendendo a relação



→ Cada gafanhoto pode assistir vários cursos

→ Cada curso pode ser assistido por várias pessoas

Inserindo os registros

```
INSERT INTO gafanhoto_assiste_curso  
(DEFAULT, '2014-03-01', '1', '2');
```

Aqui eu dico que meu **id** será gerado automaticamente

data em que assistiu

idgafanhoto

idcurso

```
insert into gafanhotos_assiste_curso values  
(default, '2014-03-01', '1', '2');
```

id	data	idgafanhoto	idcurso
1	2014-03-01	1	2
2	2015-12-22	3	6
3	2014-01-01	22	12
4	2016-05-12	1	19
NULL	NULL	NULL	NULL

Os outros foram gerados manualmente

→ E se eu quiser fazer aparecer o nome do gafanhoto e o nome do curso? Lembrando que o gafanhoto está de um lado, o curso do outro e eu tenho uma tabela no meio

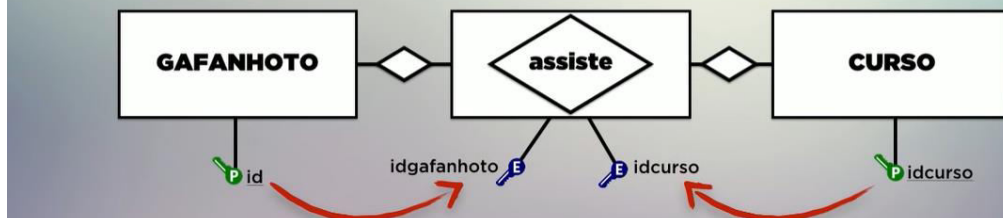
Junções

```
SELECT * FROM gafanhotos g  
JOIN gafanhoto_assiste_curso a  
ON g.id = a.idgafanhoto  
JOIN cursos c  
ON a.idcurso = c.idcurso;
```

Aqui eu apelidei a tabela **gafanhotos** como **g** e a tabela **gafanhoto_assiste_curso** como **a**

JOIN de

JOIN de **curso**



```
SELECT  
g.nome, c.nome  
FROM  
gafanhotos g  
JOIN  
gafanhotos_assiste_curso a ON g.id = a.idgafanhoto  
JOIN  
cursos c ON a.idcurso = c.idcurso  
ORDER BY g.nome;
```

Aqui eu peço pra mostrar o nome que vem da tabela de **gafanhotos**

Aqui eu peço para mostrar o nome que vem da tabela de **cursos**

nome	nome
Daniel Moraes	Algoritmos
Daniel Moraes	Redes
Emerson Gabriel	MySQL
Guilherme de Sousa	C++