

# Complementos de Bases de Dados 2020/2021

Licenciatura em Eng<sup>a</sup>. Informática

Entrega Final - Relatório Técnico

Turma: SW02

Horário de Laboratório: 5<sup>a</sup> feira 13h00

Docente: João Santos

Grupo:

Nº180221011, Tiago Farinha

Nº180221015, Francisco Moura

### 1. Sumário Executivo

Este relatório representa a documentação de análise, levantamento de requisitos e desenvolvimento de um projeto no âmbito da unidade curricular de Complementos de Bases de Dados.

O projeto consiste na reestruturação de uma base de dados. A nova base de dados será intitulada de AgrupamentoSTB e, a antiga, de OldSTB. Serão apresentados os requisitos funcionais da nova base de dados, assim como, as entidades que nela atuam e o modelo relacional. Além disso, serão também indicados procedures, novos dados etc. aplicados à nova base de dados.

### 2. Especificação de Requisitos

#### 2.1 Requisitos funcionais

ID	Descrição	Implementado (S/N)
RF01	<i>O sistema deverá permitir a um aluno/encarregado de educação criar uma conta e codificar a sua Password</i>	S
RF02	O sistema deverá permitir alterar a password de um utilizador	S
RF03	O sistema deverá permitir fazer a autenticação	+-
RF04	O sistema deverá permitir adicionar um novo aluno	S
RF05	O sistema deverá permitir adicionar um novo Encarregado de Educação	S
RF06	O sistema deverá permitir registar as inscrições nas disciplinas de cada aluno para o ano letivo corrente	S
RF07	O sistema deverá permitir a gestão dos anos letivos	S
RF08	O sistema deverá permitir gerir os utilizadores e os seus acessos	+-

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

RF09	O sistema deverá permitir adicionar dados multi-idioma	+-
RF10	O sistema deverá permitir alterar as notas dos alunos num determinado Período	S
RF11	O sistema deverá permitir inscrever um aluno numa determinada disciplina	S
RF12	O sistema deverá permitir a abertura de um novo ano letivo	S
RF13	O sistema deverá permitir o fecho do ano letivo corrente	S
RF14	O sistema deverá permitir o cálculo das médias das notas e a sua taxa de crescimento relativamente a anos anteriores	S
RF15	O sistema deverá permitir devolver uma mensagem de erro “amigável” aquando da execução de SPs, Functions etc com dados incorretos.	S
RF16	O sistema deverá permitir o envio de um email aquando da alteração de password de um utilizador	S
RF17	O sistema deverá permitir que sejam gerados 60 anos de dados (Índices)	S
RF18	O sistema deverá permitir calcular a taxa de crescimento de cada ano, face ao ano anterior, no que ao número de alunos diz respeito.	S
RF19	O sistema deverá permitir calcular, por ano, a percentagem de alunos com nota final (P3) maior ou igual a 15	S
RF20	O sistema deverá permitir calcular, por cada ano, a escola com melhor média final	S
RF21	O sistema deverá permitir a realização de backups integrais, diferenciais e de logs	S
RF22	O sistema deverá permitir a realização do backup do tail dos logs	S
RF23	O sistema deverá permitir recuperar a base de dados em caso de falhas.	S
RF24	O sistema deverá permitir aos utilizadores aceder aos dados que lhes são devidos.	S

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

RF25	O sistema deverá permitir executar concorrencialmente diversas transações, sem causar erros (Dirty-Read, Phantom Read etc)	S
------	--	---

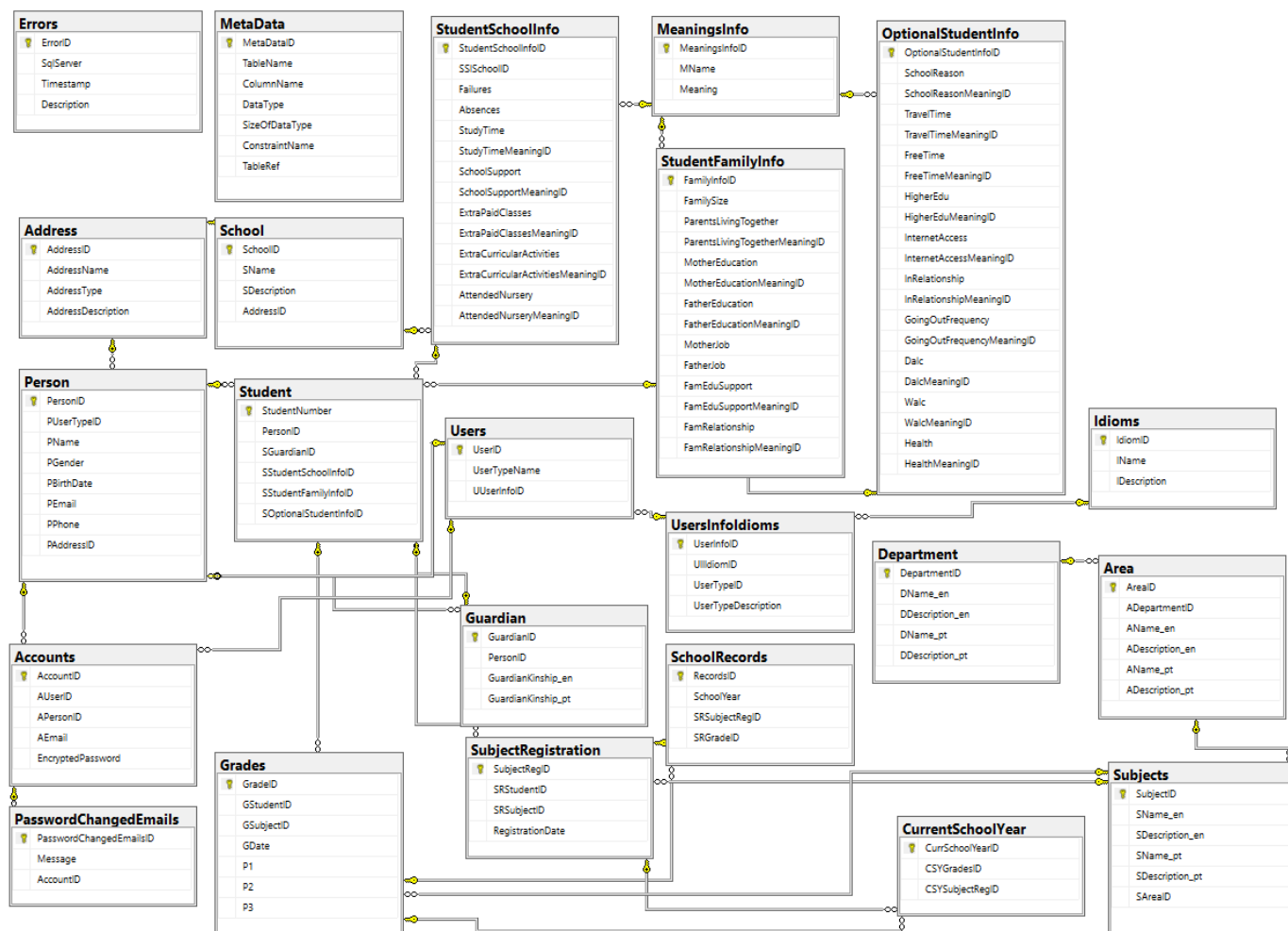
### 2.2Atores

Nome	Descrição
Users	UserID (PK), UserType
Accounts	AccountID (PK), AEmail, EncryptedPassword, AUserID (FK)
Address	AddressID (PK), AddressName, AddressType, AddressDescription
Department	DepartmentID (PK), DName_en, DDescription_en , DName_pt , DDescription_pt
Area	AreaID (PK), DepartmentID (FK), AName_en, ADescription_en, AName_pt, ADescription_pt
Subjects	SubjectID (PK), SName_en, SDescription_en, SName_pt, SDescription_pt, SAreaID (FK)
Guardian	GuardianID (PK), PersonID (FK), GuardianKinship_en, GuardianKinship_pt
Idioms	IdiomID (PK), IName, IDescription
MeaningsInfo	MeaningsInfoID (PK), MName, Meaning
Student	StudentNumber (PK), PersonID (FK), SGuardianID (FK), SStudentSchoolInfoID (FK), SStudentFamilyInfoID (FK), SOptionalStudentInfoID (FK)
Person	PersonID (PK), PersonRole_en, PersonRole_pt, PName, PGender, PBirthdate, PEmail, PPhone, PAddressID (FK)
School	SchoolID (PK), SName, SDescription, AddressID (FK)

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

StudentSchoolInfo	StudentSchoolInfoID, SSISchoolID (FK), Failures, Absences, StudyTime, SchoolSupport, ExtraPaidClasses, ExtraCurricularActivities, AttendedNursery
StudentFamilyInfo	FamilyInfoID (PK), FamilySize, ParentsLivingTogether, MotherEducation, FatherEducation, MotherJob, FatherJob, FamEduSupport, FamRelationship
OptionalStudentInfo	OptionalStudentInfo (PK), SchoolReason, TravelTime, FreeTime, HigherEdu, InternetAccess, InRelationship, GoingOutFrequency, Dalc, Walc, Health
SubjectRegistration	SubjectRegID (PK), SRStudentID (FK), SRSubjectID (FK), RegistrationDate
Grades	GradeID (PK), GStudentID (FK), GSubjectID (FK), P1, P2, P3
SchoolRecords	RecordsID (PK), SchoolYear, SRSubjectRegID (FK), SRGradeID (FK)
CurrentSchoolYear	CurrSchoolYearID (PK), CSYGradesID (FK), CSYSubjectRegID (FK)
Erros	ErroID, SqlServer, Timestamp, Description
Metadata	MetadataID, TableName, ColumnName, DataType, SizeOfDataType, ConstraintName, TableRef
UsersInfoIdioms	UserInfoID (PK), UIIdiomID (FK), UserTypeID, UserTypeDescription

## 1. Modelo Relacional (*Modelo de dados*)



## 2. Dimensionamento do Layout

*Descriminando a projeção de necessidades que conduz ao layout proposto*

*Identificação do espaço ocupado por tabela*

Nome Tabela	Dimensão do Registo (KB)	Nº de Registos (inicial/final)
Accounts	0	0/0
Address	128	1947/1949
Area	16	2/2
CurrentSchoolYear	16	0/1
Department	16	2/2
Errors	0	0/0
Grades	208	5841/5842
Guardian	80	1947/1947
MetaData	16	0/111
OptionalStudentInfo	112	1947/1947
PasswordChangedEmails	0	0/0
Person	480	3894/3894
School	16	2/2
SchoolRecords	168	5841/5841

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

Student	80	1947/1947
StudentFamilyInfo	112	1947/1947
StudentSchoolInfo	80	1947/1947
SubjectRegistration	184	0/5842
Subjects	16	3/3
Users	48	0/3

### 3. Views

Nome	Descrição
Get_NewID	<i>Esta view permite obter um novo ID aleatório.</i>
View_TaxaCrescimentoNrAlunosEmRelacaoAoAnoAnterior	<i>Esta view permite calcular a taxa de crescimento de cada ano, face ao ano anterior, no que ao número de alunos diz respeito.</i>
View_AlunosComNotaFinalMaiorQueQuinze	<i>Esta view permite calcular, por ano, a percentagem de alunos com nota final (P3) maior ou igual a 15.</i>
View_EscolaComMelhorMediaFinalPorAno	<i>Esta view permite calcular, por cada ano, a escola com melhor média final.</i>
view_KeyUserAccess	<i>Esta view permite filtrar a informação que os estudantes podem aceder (consoante o seu email).</i>



## 4. Functions

Nome	Tipo	Atributos	Descrição
randomString	Varchar(MAX)	@startingString, @counter	Permite obter uma String aleatória de caracteres
Hashing	Varchar(MAX)	@String	Permite encriptar uma String (neste caso, é usado para encriptar o email e a password dos utilizadores)
maxAvgPorAno	Table	-	Esta é uma função auxiliar que foi necessária para gerar os resultados da <code>View_EscolaComMelhorMediaFinalPorAno</code> pois, devido à forma como implementámos a nossa BD, não encontramos uma maneira melhor de encontrar estes resultados com uma simples query.

## 5. Stored procedures

Nome	Atributos	Descrição
AbrirAnoLetivo	-	Permite criar a tabela do novo ano letivo usando o catálogo de metadados.
FecharAnoLetivo	-	Permite passar os dados do ano letivo corrente para o histórico usando o catálogo de metadados
InscriverAlunoDisciplina	@StudentNumber, @SubjectID	Permite inscrever um aluno numa disciplina
AtualizarNota	@StudentNumber, @SubjectID, @Year, @Period, @NewGrade	Permite atualizar a nota de um aluno numa certa disciplina e Período.
AlunosInscritosETaxaCrescimento	-	Permite ver o total de alunos de alunos inscritos em cada uma das disciplinas no ano aberto face ao ano anterior e a respetiva taxa de crescimento.
MediaPorAnoETaxaCrescimento	@year	Permite ver a média das notas de todos os anos por escola num determinado ano e a respetiva taxa de crescimento face ao ano anterior.
insertGrade	@temp_GStudentID, @temp_GSubjectID, @temp_grade1, @temp_grade2, @temp_grade3	Permite adicionar notas a um determinado aluno numa determinada disciplina.
Autenticar	@email, @Password	Permite fazer o login de um utilizador
Terminar_Sessao	-	Permite terminar a sessão do utilizador que está com a sessão iniciada
Registar	@email, @Password, @UserType	Permite criar uma nova conta de Utilizador

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

AlterarPalavrapasse	@CurrPassword, @NewPassword1, @NewPassword2	Permite a um utilizador alterar a password
AssociarFKStudentSchoolInfoID	-	Permite associar a FK StudentSchoolInfoID à tabela Student de forma a ligar as informações escolares dos alunos pertencentes à OldSTB
InscricoesDisciplinas	-	Permite popular as inscrições dos alunos nas disciplinas dos dados da OldSTB
RecuperarPalavraPasse	@email	Permite ao utilizador recuperar a palavra passe da sua conta
AcederInformacao	-	Permite ao utilizador aceder aos seus dados
CriarAluno	@StudentID, @Name, @Email, @GuardianID	Permite a criação de um novo aluno.
CriarEncarregado	@Name, @Email, @GuardianKinship_en	Permite a criação de um novo Encarregado de Educação.
CriarDisciplina	@Name_en, @Description_en, @Name_pt, @Description_pt, @DepartmentID, @AreaID	Permite a criação de uma nova disciplina.
CriarArea	@Name_en, @Description_en, @Name_pt, @Description_pt	Permite a criação de uma nova área.
CriarDepartamento	@Name_en, @Description_en, @Name_pt, @Description_pt	Permite a criação de um novo departamento.
GerarDadosDeHistorico	-	Permite gerar dados de histórico desde o ano 1960 até 2020 para 9 disciplinas e 4 escolas (entre 2000-2500 registos por ano).
spNovoStudentUser	@studentEmail, @password	Permite criar um novo utilizador do tipo estudante.
fullBackup	-	Permite criar um backup completo.
partialBackup_1	-	Permite criar um backup diferencial.

## *2ª Fase Relatório Técnico – Complementos de Bases de Dados*

partialBackup_2	-	Permite criar um backup diferencial.
logBackup	-	Permite fazer o backup do log.
backupLogTail	-	Permite fazer o backup do tail do log.
restoreFullBackup	-	Permite restaurar o backup completo.
restorePartialBackup_1	-	Permite restaurar o backup diferencial.
restorePartialBackup_2	-	Permite restaurar o backup diferencial.
restoreLog	-	Permite restaurar o backup do log.
restoreLogTail	-	Permite restaurar o backup do tail do log.

## 6. Consultas

### 6.1 Verificação da conformidade dos dados

```
7. use AgrupamentoSTB;
8.
9.
10. -- |-----|
11.
12. -- |                2.1.3 - VERIFICAÇÃO DA NOVA BD                |
13.
14. -- |-----|
15.
16.
17. -- a) Média de notas no ano letivo por escola
18. select SchoolYear as 'Ano', SDescription as 'Escola', avg((P1 + P2 + P3)/3) as 'Média' from
    Grades g
19. inner join Student s on g.GStudentID = s.StudentNumber
20. inner join SchoolRecords sr on g.GradeID = sr.SRGradeID
21. inner join StudentSchoolInfo ssi on s.SStudentSchoolInfoID = ssi.StudentSchoolInfoID
22. inner join School sc on ssi.SSISchoolID = sc.SchoolID
23. group by SchoolYear, SDescription order by SchoolYear;
24.
25. select * from Student
26.
27. -- -----
28.
29. -- b) Média de notas por ano letivo e período letivo por escola
30. select SchoolYear as 'Ano', SDescription as 'Escola', avg(P1) as 'Média P1', avg(P2) as
    'Média P2', avg(P3) as 'Média P3' from Grades g
31. inner join Student s on g.GStudentID = s.StudentNumber
32. inner join SchoolRecords sr on g.GradeID = sr.SRGradeID
33. inner join StudentSchoolInfo ssi on s.SStudentSchoolInfoID = ssi.StudentSchoolInfoID
34. inner join School sc on ssi.SSISchoolID = sc.SchoolID
35. group by SchoolYear, SDescription;
36.
37.
38. -- -----
39.
40. -- c) Total de alunos por escola/ano letivo
41. select SchoolYear as 'Ano', SDescription as 'Escola', COUNT(StudentNumber) / 3 as 'Número de
    alunos'
42. from SchoolRecords sr
43. inner join Grades g on sr.SRGradeID = g.GradeID
44. inner join Student s on g.GStudentID = s.StudentNumber
45. inner join StudentSchoolInfo ssi on s.SStudentSchoolInfoID = ssi.StudentSchoolInfoID
46. inner join School sc on ssi.SSISchoolID = sc.SchoolID
47. group by SchoolYear, SDescription;
48.
49.
50.
51. -- -----
52. -- -----
53.
```

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

```
54.
55.
56. -- |-----|
57.
58. -- |                                     |
59.                                     EXTRAS/Outras Consultas
60. -- |-----|
61.
62.
63. -- OldSTB
64. use OldSTB;
65.
66. -- 1) Número de alunos rapazes da escola Mousinho da Silveira inscritos em BD nos últimos 3
67.   anos (240 alunos)
68. SELECT (SELECT COUNT([Student Number]) FROM    dbo.[2017 student-BD] where school = 'MS' and
69.   sex = 'M')
70.       +
71.       (SELECT COUNT(Student_Number) FROM    dbo.[2018 student-BD] where school = 'MS' and
72.   sex = 'M')
73.       +
74.       (SELECT COUNT(Student_Number) FROM    dbo.[2019 student-BD] where school = 'MS'
75.   and sex = 'M'
76.   ) AS 'Total de alunos rapazes da Mousinho Silveira'
77.
78.
79. -- 2) Número de alunos com notas iguais ou superiores a 10 no 1º Período
80. select (SELECT COUNT([Student Number]) FROM dbo.[2017 student-BD] where P1 >= 10)
81.       +
82.       (SELECT COUNT(Student_Number) FROM dbo.[2018 student-BD] where P1 >= 10)
83.       +
84.       (SELECT COUNT(Student_Number) FROM dbo.[2019 student-BD] where P1 >= 10
85.       )
86.       +
87.       (SELECT COUNT(Student_Number) FROM dbo.[2017 student-CBD] where P1 >= 10
88.       )
89.       +
90.       (SELECT COUNT(Student_Number) FROM dbo.[2018 student-CBD] where P1 >= 10
91.       )
92.       +
93.       (SELECT COUNT(Student_Number) FROM dbo.[2019 student-CBD] where P1 >= 10
94.       )
95.       +
96.       (SELECT COUNT(Student_Number) FROM dbo.[2017 student-MAT1] where P1 >= 10
97.       )
98.       +
99.       (SELECT COUNT(Student_Number) FROM dbo.[2018 student-MAT1] where P1 >= 10
100.      )
101.      +
102.      (SELECT COUNT(Student_Number) FROM dbo.[2019 student-MAT1] where P1 >= 10
103.      ) AS 'Total de alunos com nota >= 10 no P1'
```

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

```
103.      -- 3) Dados de um aluno específico
104.      select [Student Number], school, sex, [Birth Date], guardian from dbo.[2017 student-
BD] where [Student Number] = 1;
105.
106.      -- 4) Disciplinas, Áreas e Departamentos
107.      select (select distinct Class from dbo.[2017 student-BD]) union
108.      (select distinct Class from dbo.[2017 student-CBD]) union
109.      (select distinct Class from dbo.[2017 student-MAT1]);
110.
111.      -- 5) Número de chumbos de um aluno ao acaso
112.      select [Student Number] as 'Número de Estudante', convert(int, a.failures) +
convert(int, b.failures) + convert(int, c.failures) as 'Nº de Chumbos' from dbo.[2017
student-BD] a join [dbo].[2017 student-CBD] b on a.[Student Number] = b.Student_Number
113.      join dbo.[2017 student-MAT1] c on b.Student_Number = c.Student_Number where [Student
Number] = 164;
114.
115.      -----
-----
116.
117.
118.
119.      -- AgrupamentoSTB
120.      use AgrupamentoSTB;
121.
122.      -- 1) Número de alunos rapazes da escola Mousinho da Silveira inscritos em BD nos
últimos 3 anos (240 alunos)
123.      select count(*) as 'Total de alunos rapazes da Mousinho Silveira' from Person p
124.      inner join Student s on p.PersonID = s.PersonID
125.      inner join StudentSchoolInfo ssi on s.SStudentSchoolInfoID = ssi.StudentSchoolInfoID
126.      inner join School sc on ssi.SSISchoolID = SchoolID
127.      where p.PGender = 'M' and sc.SName = 'MS';
128.
129.
130.
131.      -- 2) Número de alunos com notas iguais ou superiores a 10 no 1º Período
132.      select count(P1) from Grades where P1 >= 10;
133.
134.
135.
136.      -- 3) Dados de um aluno específico
137.      select StudentNumber, SName, PGender, PBirthDate, GuardianKinship_en from Student s
138.      inner join Person p on s.PersonID = p.PersonID
139.      inner join StudentSchoolInfo ssi on s.SStudentSchoolInfoID = ssi.StudentSchoolInfoID
140.      inner join School sc on ssi.SSISchoolID = sc.SchoolID
141.      inner join Guardian g on s.SGuardianID = g.GuardianID
142.      where StudentNumber = 1;
143.
144.
145.
146.      -- 4) Disciplinas, Áreas e Departamentos
147.      select SName_pt as UC, SDescription_pt as 'Descrição da UC', AName_pt as 'Área',
ADescription_pt as 'Descrição da Área',
148.      DName_pt as Departamento, DDescription_pt as 'Descrição do Departamento'
```

## 2ª Fase Relatório Técnico – Complementos de Bases de Dados

```
149.      from Subjects s inner join Area a on s.SAreaID = a.AreaID
150.      inner join Department d on s.SDepartmentID = DepartmentID;
151.
152.
153.      -- 5) Número de chumbos de um aluno ao acaso
154.      select StudentNumber, Failures from Student s inner join StudentSchoolInfo ssi on
      s.SStudentSchoolInfoID = ssi.StudentSchoolInfoID
155.      where StudentNumber = 164;
156.
157.      select * from Student;
```

### 157.1 Outras consultas

```
select * from Accounts;
select * from Address;
select * from Area;
select * from CurrentSchoolYear;
select * from Department;
select * from Errors;
select * from Grades;
select * from Guardian;
Select * from Idioms;
Select * from MeaningsInfo;
select * from MetaData;
select * from OptionalStudentInfo;
select * from PasswordChangedEmails;
select * from Person;
select * from School;
select * from SchoolRecords;
select * from Student;
select * from StudentFamilyInfo;
select * from StudentSchoolInfo;
select * from SubjectRegistration;
select * from Subjects;
select * from Users;
select * from UsersInfoIdioms;
```



### 7. Descrição da Demonstração

Para o correto funcionamento da nova base de dados (AgrupamentoSTB) é necessário correr os scripts na seguinte ordem:

1. DataBaseCreation\_Layout;
2. CREATETabelasNovaBD;
3. Migração de Dados;
4. NewData;
5. MetaData;
6. ScriptVerificaçãoDaBD;
7. ScriptGestãoUtilizadores;
8. ScriptStoredProcedures;
9. Índices;
10. Backup&Recuperação;
11. NíveisAcesso;
12. Transações;
13. MongoDB;

### 8. Melhorias em relação à Fase 1

Para a Fase 2 do projeto, decidimos melhorar os aspetos apontados pelo professor.

O primeiro ponto foi a falta de uma ligação direta entre a tabela Department e a tabela Area (hierarquia), estando estes apenas ligados pela tabela Subjects. Ora, de forma a fortalecer as associações entre as tabelas, alterámos a estrutura de forma a que o Department esteja ligado à Area por uma FK e, por sua vez, a Area liga-se ao Subjects pela respetiva FK.

O segundo ponto foi a má implementação de suporte multi-idioma. Para resolver este problema, criámos uma tabela de idiomas e tabelas intermediárias que contêm apenas as informações em que seja necessário traduzir.

Devido a falta de tempo, apenas fizemos estas traduções para uma tabela (Users e UsersInfoldioms), de forma a demonstrar a implementação melhorada deste requisito.

O terceiro e último ponto foi a inexistência dos significados de vários parâmetros das tabelas OptionalStudentInfo, StudentFamilyInfo e StudentSchoolInfo (por exemplo: travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)). Para resolver isto, criámos uma tabela com todos os significados possíveis e, depois, associámos às diversas FKs criadas nestas tabelas, criando então uma associação dos parâmetros com os seus significados.

## 9. Índices

Nome	Tabela	Justificação/Consultas
<b>Ix_SchoolRecords1</b>	<b>SchoolRecords</b>	Permite baixar de 5792 para 4697 logical reads na execução da View_TaxaCrescimentoNrAlunosEmRelacaoAoAnoAnterior
<b>Ix_SchoolRecords2</b>	<b>SchoolRecords</b>	Permite baixar de 4697 para 4331 logical reads na execução da View_TaxaCrescimentoNrAlunosEmRelacaoAoAnoAnterior
<b>Ix_Grades</b>	<b>Grades</b>	Permite baixar de 745 para 742 logical reads na execução da View_TaxaCrescimentoNrAlunosEmRelacaoAoAnoAnterior
<b>Ix_Student</b>	<b>Student</b>	Permite baixar de 585 para 517 logical reads na execução da View_TaxaCrescimentoNrAlunosEmRelacaoAoAnoAnterior
<b>[_dta_index_Grades_7_9016_82360__K7_4]</b>	<b>Grades</b>	Permite baixar de 45136 para 30866 logical reads na execução da View_AlunosComNotaFinalMaiorQueQuinze
<b>[_dta_index_SchoolRecords_7_965682588__K2]</b>	<b>SchoolRecords</b>	Permite baixar de 1327 para 748 logical reads na execução da View_AlunosComNotaFinalMaiorQueQuinze

A view **View\_EscolaComMelhorMediaFinalPorAno** não precisou do uso de índices, pois foi utilizada uma função auxiliar (**maxAvgPorAno()**) que já calcula os dados pretendidos, otimizando a procura e cálculo dos resultados. Implementámos a view desta forma, pois utilizando apenas o select para filtrar os resultados, era-nos devolvido as médias de todas as escolas por ano e não apenas a média mais alta por ano.

Para a construção e melhoramento da pesquisa, criámos os índices baseados na ferramenta “Database Engine Tuning Advisor”.

Com esta ferramenta, verificámos quais os índices recomendados e escolhemos os mais indicados, tendo como ponto de decisão o número de logical reads. Quando verificámos que havia um número significativo na redução de logical reads, implementámos então os índices.

Com o uso de índices, reparámos também que as execuções das consultas têm um tempo de execução menor.

Comparando planos de execução, constatámos que o custo das operações estava mais disperso pelos diversos índices e operações do tipo join (por exemplo), reduzindo o custo que previamente pesava maioritariamente no spool índice (The **Index Spool** operator is one of the four spool operators that SQL Server supports. It retains a copy of all data it reads in an indexed worktable (in tempdb), and can then later return subsets of these rows without having to call its child operators to produce them again.).

## 10. MongoDB

Para podermos popular a nova Base de Dados em MongoDB, decidimos optar pela exportação para ficheiros CSV.

Para isso, seguimos as seguintes etapas:

- 1º: Botão direito do rato na BD AgrupamentoSTB -> Tasks -> Export Data.
- 2º: Data Source -> SQL Server Native Client 11.0.
- 3º: Destination -> Flat File Destination.
- 4º: Selecionar o ficheiro em formato .csv (previamente criado sem nada).
- 5º: Selecionar a opção 'Write a query to specify the data to transfer' (e aqui escolhemos as colunas que pretendemos através de uma query).
- 6º: Next -> Next (mudar o column delimiter para semicolon (;)) -> Run Immediately.

Após este processo, escolhemos como IDE de MongoDB o **NoSQLBooster for MongoDB** e procedemos à criação da base de dados e das suas respetivas collections.

Para isso desenvolvemos o Script **CreateBD&Collections**, o qual apenas tem o comando de criação da BD e das Collections necessárias ao que foi pedido no enunciado.

Depois, apenas tivemos de importar os ficheiros .csv para fazer a devida população da nova BD (**STBWeb**).

Após a importação, procedemos ao desenvolvimento dos comandos que cumpram o que foi requisitado no enunciado do projeto (Script **Comandos (ponto 5 do enunciado)**).



### 11. Encriptação

Para a encriptação da palavra passe e email dos utilizadores, utilizámos uma função de hashing **SHA2\_512** pois estes dados não devem ser desencriptados. A função de hashing apenas funciona num sentido, facilitando a manutenção dos dados encriptados.

Nota: A Encriptação foi feita já na 1ª fase, no Script ScriptGestaoUtilizadores.

### 12. Backup e Restore

Implementámos quatro SPs para backups. Os backups consistem num backup completo, de doze em doze horas para um HDD, dois backups diferenciais, sendo de duas em duas horas alternadamente (primeiro o backup diferencial 1, seguido do backup diferencial 2) para o mesmo HDD que o backup completo e o backup dos logs de quinze em quinze minutos para um HDD diferente do anterior.

O restore são seis SPs. Deve-se começar por fazer um backup do tail dos logs com no\_truncate, pois permite maximizar a recuperação do tail dos logs. Será depois restaurado o backup completo, seguido do backup parcial mais antigo e depois o backup parcial mais recente. Termina-se com o restauro dos logs seguido do restauro do tail dos logs, este tendo de ser obrigatoriamente o último a ser restaurado.

Backup Completo -> Backup Diferencial 1 -> CRASH -> Backup Logs Tail -> Restore Completo -> Restore Diferencial 1 -> Restore do último backup de logs -> Restore Logs Tail

Backup Completo -> Backup Diferencial 1 -> Backup Diferencial 2 -> Backup Completo -> CRASH -> Backup Logs Tail -> Restore Completo -> Restore do último backup de logs -> Restore Logs Tail

No caso de certas tabelas raramente serem alteradas podem-se mover para um file group distinto e fazer backup dos restantes file groups com uma frequência mais elevada do que os backups deste novo file group.

### 13. Transações e Concorrência

Decidimos usar o nível de isolamento **Repeatable Read** porque este permite o acesso a dados, mesmo que estes estejam a ser concorrentemente manipulados/criados.

Outra razão foi o facto o **Repeatable Read** garantir que é devolvido o resultado do 1º **Select**, mesmo que haja alterações aos dados, mas não impedindo que estes resultados sejam então depois alterados (ficam bloqueados durante a transação de consulta).

Por exemplo, ao fazermos um **delete** da Tabela SchoolRecords (T1) (todos os registos de 2017, por exemplo) numa transação e tentarmos aceder aos dados das notas de um aluno associadas a esse ano (T2) simultâneamente, o utilizador que está a tentar aceder a esses dados (T2), consegue ainda consultá-los.

Isto foi também útil em transações em que se fechou o ano letivo (T1), pois evitou uma mensagem de erro e mostrou ainda os resultados do ano em questão (T2) (sem o utilizador ter o trabalho extra de iniciar uma nova consulta ao SchoolRecords).

*Usámos também o **Read Committed** no Cenário 2 para consultas em que novos dados eram concorrentemente inseridos, pois assim conseguimos aceder aos dados antes da inserção, não sendo afetados por esta, e permitindo a escrita dos novos dados.*

### 14. Níveis de Acesso

Neste módulo, criámos utilizadores e os seus roles. Depois, definimos os privilégios que cada tipo de utilizador (Role) tem direito.

Criámos também testes (consultas), de forma a validar estas restrições.



## 15. Conclusões

Podemos concluir, com a finalização deste projeto, que cumprimos a maioria dos requisitos pedidos de forma relativamente eficiente e de forma efetivamente rápida.

Certos aspetos não foram implementados da melhor forma, mas ganhámos bons conhecimentos de Bases de dados ao longo do semestre que nos permitiram fazer o nosso melhor e nos ajudarão no nosso futuro profissional.