```
CREATE TABLE BANK(
        bank_code INTEGER,
        bank_name VARCHAR(40),
        bank_headquater VARCHAR(40),
        CONSTRAINT bank_code_pk PRIMARY KEY(bank_code)

);

-- KEEPS DIFFERENT BRANCHES OF A BANK
CREATE TABLE BANKBRANCH(
        bank_branch_code INTEGER,
        bank_code INTEGER,
        bank_branch_address VARCHAR(40),
        bank_branch_phone VARCHAR(40),
        bank_branch_email VARCHAR(40),
        CONSTRAINT bank_pk PRIMARY KEY(bank_branch_code),
        CONSTRAINT bankbranch_bank_fk FOREIGN KEY(bank_code) REFERENCES BANK(bank_code)
);

-- DIFFERENT QUALIFICATIONS THAT MIGHT BE POSSESSED BY A CONTRACTOR
CREATE TABLE QUALIFICATION(
        qualification_code INTEGER IDENTITY(1,1),
        qualification_name  VARCHAR(40),
        CONSTRAINT qualification_code_pk PRIMARY KEY(qualification_code)
);

CREATE TABLE REGION (
 region_code INTEGER,
```

```
 region_name VARCHAR(40),
 CONSTRAINT region_pk PRIMARY KEY (region_code)
);

-- DIFFERENT TYPES OF DOCUMENTS THAT MIGHT BE PRESENTED FOR IDENTIFICATION BY CONTRACTORS
CREATE TABLE DOCUMENT(
        document_code INTEGER IDENTITY(100,1),
        document_name VARCHAR(40),
        CONSTRAINT document_pk PRIMARY KEY(document_code)
);

-- KEEPS ALL THE CONTRACT TYPES THAT ARE AVAILABE IN WAEC
CREATE TABLE CONTRACT (
        contract_code INTEGER IDENTITY(1,1),
        contract_type VARCHAR(40),
        CONSTRAINT contract_pk PRIMARY KEY(contract_code)
);

CREATE TABLE SCHOOL(
        scl_code VARCHAR(10),
        scl_name VARCHAR(40),
        sch_level VARCHAR(40),
        scl_region INTEGER,
        CONSTRAINT junsch_pk PRIMARY KEY(scl_code),
        CONSTRAINT sch_region_fk FOREIGN KEY(scl_region) REFERENCES REGION(region_code)
);

CREATE TABLE CONTRACTOR(
```

```
	contractor_code INTEGER,
	contract_code INTEGER,
	first_name VARCHAR(40),
	middle_name VARCHAR(40),
	last_name VARCHAR(40),
	dob DATE,
	photo VARBINARY(MAX),
	registration_date DATE ,
	nationality VARCHAR(40),
	document INTEGER,
	phone VARCHAR(20), -- newly added
	address VARCHAR(20), -- newly added
	status VARCHAR(40), --ACTIVE OR INACTIVE
	field_of_study VARCHAR(40),
	contractor_region INTEGER,
	CONSTRAINT contractor_pk PRIMARY KEY(contractor_code),
	CONSTRAINT contract_detail_fk FOREIGN KEY(contract_code) REFERENCES
CONTRACT(contract_code),
	CONSTRAINT contractor_document_fk FOREIGN KEY(document) REFERENCES
DOCUMENT(document_code),
	CONSTRAINT contractor_region_fk FOREIGN KEY(contractor_region) REFERENCES
REGION(region_code)
);

CREATE TABLE CONTRACTORBANK(
	contractor_bank INTEGER,
	contractor INTEGER,
	account_Number VARCHAR(40),
	account_name VARCHAR(40),
	bban VARCHAR(40),
```

```
	CONSTRAINT contrabank_pk PRIMARY KEY(contractor_bank, contractor, account_Number),

	CONSTRAINT contractor_bank_fk FOREIGN KEY(contractor_bank) REFERENCES
BANK(bank_code),

	CONSTRAINT contrabank_contractor_INTEGER_fk FOREIGN KEY(contractor) REFERENCES
CONTRACTOR(contractor_code)

);


--KEEPS CONTRACTOR QUALIFICATIONS

CREATE TABLE CONTRACTORQUALIFICATION(

	contra_qual_code INTEGER,

	contractor_id INTEGER,

	qualification_date DATE,

	description VARCHAR(300),

	CONSTRAINT contractor_qual_pk PRIMARY KEY(contra_qual_code, contractor_id),

	CONSTRAINT contra_qual_fk FOREIGN KEY(contra_qual_code) REFERENCES
QUALIFICATION(qualification_code),

	CONSTRAINT contractor_id_fk FOREIGN KEY(contractor_id) REFERENCES
CONTRACTOR(contractor_code)

);


CREATE TABLE CONTRACTORSCHOOL(

	contractor_code INTEGER,

	contractor_sch VARCHAR(10),

	contracotr_subject VARCHAR(10), -- newly added

	join_date DATE,

	status VARCHAR(20), -- this indentified whether the school is current or previous

	CONSTRAINT contractor_school_pk PRIMARY KEY(contractor_code, contractor_sch),

	CONSTRAINT contractor_school_fk FOREIGN KEY(contractor_sch) REFERENCES
SCHOOL(scl_code),

	CONSTRAINT contractor_code_fk FOREIGN KEY(contractor_code) REFERENCES
CONTRACTOR(contractor_code)
```

```
);

CREATE TABLE CONTRACTORTUTORIALCLASS(
        contractorID INTEGER,
        schoolID VARCHAR(10),
        start_date DATE,
        status VARCHAR(10), -- whether current or previous
        CONSTRAINT tutorial_class_pk PRIMARY KEY(contractorID, schoolID),
        CONSTRAINT tutorial_class_contractor FOREIGN KEY(contractorID) REFERENCES
CONTRACTOR(contractor_code),
        CONSTRAINT tutorial_class_sch FOREIGN KEY(schoolID) REFERENCES SCHOOL(scl_code)
);

CREATE TABLE EXAMDIET(
        diet_code VARCHAR(10),
        dietName VARCHAR(20),
        CONSTRAINT examDiet_dietCode_pk PRIMARY KEY(diet_code),
);

CREATE TABLE EXAMCATEGORY(
        exam_category_code VARCHAR(10),
        exam_category_name VARCHAR(100), -- WASSCE, GABECE OR NAT
        diet_code VARCHAR(10), -- INDICATES THE TIME OF THE EXAM
        CONSTRAINT examCategory_pk PRIMARY KEY(exam_category_code),
        CONSTRAINT examCategoryDiet_fk FOREIGN KEY(diet_code) REFERENCES
EXAMDIET(diet_code),
);

CREATE TABLE BLACKLIST(
```

```
	contractor_code INTEGER,
	dates DATE,
	description VARCHAR(500),
	CONSTRAINT blacklist_examiner_fk FOREIGN KEY(contractor_code) REFERENCES
CONTRACTOR(contractor_code)
);

--NOW I AM COMING TO SPECIFY TABLES FOR SUBJECTS AND EXAMS

CREATE TABLE SUBJECT(
	subject_code VARCHAR(10),
	subject_name VARCHAR(60),
	exam_category_code VARCHAR(10),
	iscore_subject VARCHAR(5),
	CONSTRAINT cabecesubject_pk PRIMARY KEY(subject_code, exam_category_code),
	CONSTRAINT SUBJECT_exam_category_code_fk FOREIGN KEY(exam_category_code)
REFERENCES EXAMCATEGORY(exam_category_code)
);

-- THIS TABLE RECORDS ALL THE SUBJECT PAPERS AND THEIR RATES
CREATE TABLE SUBJECTPAPER(
	subject_paper_code VARCHAR(10), -- THEORY, ESSAY ETC
	subject_code VARCHAR(10), -- THE EXAM SUBJECT
	exam_category_code VARCHAR(10), -- SUBJECT LEVEL (GABECE, WASSCE OR NAT)
	subject_paper_rate DECIMAL(8,2),
	CONSTRAINT gsub_paper_pk PRIMARY KEY(subject_paper_code),
	CONSTRAINT gsub_paper_fk FOREIGN KEY(subject_code, exam_category_code) REFERENCES
SUBJECT(subject_code, exam_category_code),
);
```

```
CREATE TABLE COURSEWORK(
        subject_id VARCHAR(10),
        Cwrok_exam_category VARCHAR(10), --THIS INDICATE IF PRACTICAL IS FOR WASSCE OR GABECE
        Cwork_examDiet VARCHAR(10),
        Cwork_name VARCHAR(20),
        courseword_rate DECIMAL(8,2),
        remarks VARCHAR(500),
        CONSTRAINT coursework_pk PRIMARY KEY(subject_id),
        CONSTRAINT coursework_subject_fk FOREIGN KEY(subject_id, Cwrok_exam_category)
REFERENCES SUBJECT(subject_code, exam_category_code),
        CONSTRAINT courseWork_SubjectLevel_fk FOREIGN KEY(Cwrok_exam_category) REFERENCES
EXAMCATEGORY(exam_category_code),
        CONSTRAINT courseWork_ExamDiet_fk FOREIGN KEY(Cwork_examDiet) REFERENCES
EXAMDIET(diet_code),
);


CREATE TABLE PRACTICAL(
        subject_id VARCHAR(10),
        exam_category_code VARCHAR(10), --THIS INDICATE IF PRACTICAL IS FOR WASSCE OR GABECE
        diet_code VARCHAR(10), --IDENTIFIES (MAY/JUNE OR NOVEMBER/DECEMBER)
        practical_name VARCHAR(20),
        practical_rate DECIMAL(8,2),
        remarks VARCHAR(500),
        CONSTRAINT practical_work_pk PRIMARY KEY(subject_id),
        CONSTRAINT practical_dietCode_fk FOREIGN KEY(diet_code) REFERENCES
EXAMDIET(diet_code),
        CONSTRAINT practical_work_subject_fk FOREIGN KEY(subject_id, exam_category_code)
REFERENCES SUBJECT(subject_code, exam_category_code)
);
```

```
CREATE TABLE ORAL(
        oral_subject VARCHAR(10),
        oral_exam_category VARCHAR(10),
        oral_exam_diet VARCHAR(10),
        oral_description VARCHAR(50),
        CONSTRAINT oral_oralSubject_pk PRIMARY KEY(oral_subject),
        CONSTRAINT oralSubject_subject FOREIGN KEY(oral_subject, oral_exam_category) REFERENCES
SUBJECT(subject_code, exam_category_code),
        CONSTRAINT oral_category_fk FOREIGN KEY(oral_exam_category) REFERENCES
EXAMCATEGORY(exam_category_code),
        CONSTRAINT oral_OralExamDiet_fk FOREIGN KEY(oral_exam_diet) REFERENCES
EXAMDIET(diet_code),
);

--ALLOCATE COURSEWORK TO A CONTRACTOR
CREATE TABLE COURSEWORKALLOCATION(
        SN INTEGER,
        contractor_code INTEGER,
        Csubject_code VARCHAR(10),
        number_of_works INTEGER,
        dates DATE,
        CONSTRAINT courseWorkAllocation_SN_pk PRIMARY KEY(SN),
        CONSTRAINT courseWorkAllocation_Contractor_fk FOREIGN KEY(contractor_code) REFERENCES
CONTRACTOR(contractor_code),
        CONSTRAINT courseWork_courseName_fk FOREIGN KEY(Csubject_code) REFERENCES
COURSEWORK(subject_id),
);

--ALLOCATE PRACTICAL TO A CONTRACTOR
```

```
CREATE TABLE PRACTICALALLOCATION(
        SN INTEGER,
        contractor_code INTEGER,
        practical_subject VARCHAR(10),
        number_of_works INTEGER,
        dates DATE,
        CONSTRAINT practicalAllocation_SN_pk PRIMARY KEY(SN),
        CONSTRAINT practicalAllocation_Contractor_fk FOREIGN KEY(contractor_code) REFERENCES
CONTRACTOR(contractor_code),
        CONSTRAINT practical_courseName_fk FOREIGN KEY(practical_subject) REFERENCES
PRACTICAL(subject_id),
);

--ALLOCATE ORAL TO A CONTRACTOR
CREATE TABLE ORALALLOCATION(
        SN INTEGER,
        contractor_code INTEGER,
        oral_subject VARCHAR(10),
        number_of_works INTEGER,
        dates DATE,
        CONSTRAINT oralAllocation_SN_pk PRIMARY KEY(SN),
        CONSTRAINT oralAllocation_Contractor_fk FOREIGN KEY(contractor_code) REFERENCES
CONTRACTOR(contractor_code),
        CONSTRAINT oral_courseName_fk FOREIGN KEY(oral_subject) REFERENCES ORAL(oral_subject),
);

CREATE TABLE EXAMCENTER(
        region_code INTEGER,
        centre_NUMBER VARCHAR(10),
        centre_name VARCHAR(40),
```

```
	centre_address VARCHAR(40),
	contact_person VARCHAR(40),
	phone VARCHAR(40),
	email VARCHAR(40),
	CONSTRAINT examcenter_pk PRIMARY KEY(centre_NUMBER),
	CONSTRAINT examcenter_region_fk FOREIGN KEY(region_code) REFERENCES
REGION(region_code),
);

-- SCRIPT ALLOCATION TO EXAMINERS FOR MARKING
CREATE TABLE SCRIPTALLOCATION(
	center_code VARCHAR(10),
	subjectCode VARCHAR(10),
	exam_category_code VARCHAR(10),
	examinerID INTEGER,
	subjectPaperCode VARCHAR(10),
	numOfScript INTEGER,
	numberOfScriptMarked INTEGER,
	allocatioction_date DATE,
	CONSTRAINT seniorScriptMarking_pk PRIMARY KEY(center_code, subjectCode, examinerID),
	CONSTRAINT schoolid_fk FOREIGN KEY(center_code) REFERENCES
EXAMCENTER(centre_NUMBER),
	CONSTRAINT subjectMarked_fk FOREIGN KEY(subjectCode, exam_category_code) REFERENCES
SUBJECT(subject_code, exam_category_code),
	CONSTRAINT Script_examiner_fk FOREIGN KEY(examinerID) REFERENCES
CONTRACTOR(contractor_code),
	CONSTRAINT ScriptAllocation_examPaperCode_fk FOREIGN KEY(subjectPaperCode)
REFERENCES SUBJECTPAPER(subject_paper_code),
);
```

```
-- ALLOWANCE PAYMENT TO DIFFERENT CONTRACTORS BASED ON THE LOCATION AND STATUS

CREATE TABLE TOWNSTATUS(
        status_code INTEGER,
        town_description VARCHAR(40),
        CONSTRAINT townstatus_pk PRIMARY KEY(status_code)
);

CREATE TABLE TOWN(
        town_code INTEGER,
        town_name VARCHAR(40),
        town_location INTEGER,
        CONSTRAINT town_town_code_pk PRIMARY KEY(town_code),
        CONSTRAINT townLocation_fk FOREIGN KEY(town_location) REFERENCES
TOWNSTATUS(status_code),
);

CREATE TABLE VEHICLE(
        vehicle_code INTEGER,
        vehicle_type VARCHAR(40),
        --vehicle_rate DECIMAL(8,2),
        CONSTRAINT vehicle_pk PRIMARY KEY(vehicle_code)
);

CREATE TABLE VIHECLEREGISTRATION(
        vehicle_code INTEGER,
        insurance VARCHAR(20),
        vehicleNumber VARCHAR(20),
        contractor_code INTEGER,
```

```
	CONSTRAINT VehicleRegistration_VehicleCode_Pk PRIMARY KEY(vehicleNumber),

	CONSTRAINT VihecleReg_contractor_fk FOREIGN KEY(contractor_code) REFERENCES
CONTRACTOR(contractor_code),

	CONSTRAINT VihecleReg_vehicleCode_fk FOREIGN KEY(vehicle_code) REFERENCES
VEHICLE(vehicle_code),

);


CREATE TABLE VEHICLEREFUNDRATE(

	SN INTEGER,

	townCode INTEGER,

	vehicle_code INTEGER,

	refundRate DECIMAL(8,2),

	CONSTRAINT vehiclRefundRate_towncode_pk PRIMARY KEY(townCode),

	CONSTRAINT VihecleRefundRate_towncode_fk FOREIGN KEY(townCode) REFERENCES
TOWN(town_code),

	CONSTRAINT VihecleRefundRate_vehicleCode_fk FOREIGN KEY(vehicle_code) REFERENCES
VEHICLE(vehicle_code),

);

CREATE TABLE VEHICLEREFUND(

	towncode INTEGER, -- specifies the town

	vehicle_code INTEGER,

	vehicleNumber VARCHAR(20), -- IDENTIFIES THE VEHICLE ITSELF

	contractor_code INTEGER,

	startDate DATE,

	endDate DATE,

	amountEarned DECIMAL(8,2),

	CONSTRAINT vehicle_refund_pk PRIMARY KEY(vehicleNumber),

	CONSTRAINT VihecleRefund_towncode_fk FOREIGN KEY(towncode) REFERENCES
VEHICLEREFUNDRATE(townCode),

	CONSTRAINT vehicle_code_fk FOREIGN KEY(vehicleNumber) REFERENCES
VIHECLEREGISTRATION(vehicleNumber),
```

```
        CONSTRAINT contractor_vehicleRefund_fk FOREIGN KEY(contractor_code) REFERENCES
CONTRACTOR(contractor_code),

        CONSTRAINT vehicleRefund_vehicleCode_fk FOREIGN KEY(vehicle_code) REFERENCES
VEHICLE(vehicle_code),

);


CREATE TABLE TRANSPORTFARERATE(

        towncode INTEGER,

        fareRate DECIMAL(8,2),

        CONSTRAINT transportfare_pk PRIMARY KEY(towncode, fareRate),

        CONSTRAINT TransportFare_townCode_fk FOREIGN KEY(towncode) REFERENCES
TOWN(town_code),

);


CREATE TABLE OVERNIGHTALLOWANCE(

        townCode INTEGER,

        contractor_code INTEGER,

        overnight_rate DECIMAL(8,2),

        startDate DATE,

        endDate DATE,

        amountEarned DECIMAL(8,2),

        CONSTRAINT townForOvernightAllowance_fk FOREIGN KEY(townCode) REFERENCES
TOWN(town_code),

        CONSTRAINT contractor_overnightAllowance_fk FOREIGN KEY(contractor_code) REFERENCES
CONTRACTOR(contractor_code),

);


CREATE TABLE TRANSPORTREFUND(

        townCode INTEGER,

        contractor_code INTEGER,
```

```
	startDate DATE,
	endDate DATE,
	amount_earned DECIMAL(8,2),
	CONSTRAINT TransportRefund_towncode_fk FOREIGN KEY(townCode) REFERENCES TOWN(town_code),
	CONSTRAINT TrasportRefund_contractor_code_fk FOREIGN KEY(contractor_code) REFERENCES CONTRACTOR(contractor_code),
);

--TABLES TO CAPTURE DIFFRENT MEETINGS AND ENTITLEMENT FOR EACH
CREATE TABLE MEETINGS(
	meeting_code INTEGER,
	meeting_type VARCHAR(40), -- DIFFERENT TYPES OF MEETINGS (VETTING, COORDIANTION, REPORT)
	exam_category_code VARCHAR(10), -- TALS ABOUT THE TYPE OF EXAM (GABECE, WASSCE ETC)
	meeting_rate DECIMAL(8,2),
	lunch_allowance DECIMAL(8,2),  -- LUNCH ALLOWANCE
	refreshment DECIMAL(8,2), -- REFRESHMENT ALLOWANCE
	CONSTRAINT meeting_meetingCode_pk PRIMARY KEY(meeting_code),
	CONSTRAINT meeting_exam_category_code FOREIGN KEY(exam_category_code) REFERENCES EXAMCATEGORY(exam_category_code),
);

CREATE TABLE MEETINGDETAILS( -- THIS WILL CAPTURE ALL TYPES OF MEETINGS -- VETTING AND COORDINATION MEETINGS AND REPORT
	MSN INTEGER,   -- MEETING SERIAL NUMBER
	meeting_code INTEGER,
	mStart_Date DATE,
	mEnd_date DATE,
	exam_category_code VARCHAR(10),
```

```
	subject_code VARCHAR(10),
	CONSTRAINT meetingDetails_pk PRIMARY KEY(MSN),
	CONSTRAINT meetingDetails_meetingCode_fk FOREIGN KEY(meeting_code) REFERENCES MEETINGS(meeting_code),
	CONSTRAINT meeting_examCategory_fk FOREIGN KEY(exam_category_code) REFERENCES EXAMCATEGORY(exam_category_code),
	CONSTRAINT meeting_subjectCode_fk FOREIGN KEY(subject_code, exam_category_code) REFERENCES SUBJECT (subject_code, exam_category_code),
	);

CREATE TABLE MEETINGATTENDANCE(
	meeting_code INTEGER,
	contractor_code INTEGER, -- THIS IDENTIFIED THE ID OR CODE ASSIGNED TO THE CONTRACTOR
	time_in datetime,
	time_out datetime,
	CONSTRAINT meeting_attendance_fk FOREIGN KEY(meeting_code) REFERENCES MEETINGDETAILS(MSN),
); -- NOTE: WE WILL BE ABLE TO CALCULATE THE ENTITLEMENT OF EACH PARTICIPANT AND THEIR LUNCH AND REFRESHMENT

--INVIGILATION
CREATE TABLE INVIGILATION(
	centerNo VARCHAR(10),
	invigilator_code INTEGER,
	subject_paper_code VARCHAR(10),
	invigilation_Date DATE,
	amountEarned DECIMAL(8,2),
	CONSTRAINT contractor_invigilation_fk FOREIGN KEY(invigilator_code) REFERENCES CONTRACTOR(contractor_code),
	CONSTRAINT invigilator_centerNo_fk FOREIGN KEY(centerNo) REFERENCES EXAMCENTER(centre_NUMBER),
```

```
	CONSTRAINT INVIGILATOR_subjectPaperCode_fk FOREIGN KEY(subject_paper_code) REFERENCES SUBJECTPAPER(subject_paper_code),
);

CREATE TABLE REPORTRATE(
	report_rate_code VARCHAR(10),
	report_rate DECIMAL(8,2),
	report_description VARCHAR(20),
	CONSTRAINT reportRateCode_pk PRIMARY KEY(report_rate_code),

);
CREATE TABLE REPORT(
	contractor_code INTEGER,
	diet_code VARCHAR(10),
	subject_paper_code VARCHAR(10),
	exam_category_code VARCHAR(10),
	report_rate_code VARCHAR(10),
	CONSTRAINT REPORT_contractorCode_fk FOREIGN KEY(contractor_code) REFERENCES CONTRACTOR(contractor_code),
	CONSTRAINT REPORT_deitCode_fk FOREIGN KEY(diet_code) REFERENCES EXAMDIET(diet_code),
	CONSTRAINT REPORT_subjectPaperCode_fk FOREIGN KEY(subject_paper_code) REFERENCES SUBJECTPAPER(subject_paper_code),
	CONSTRAINT REPORT_examCategory_fk FOREIGN KEY(exam_category_code) REFERENCES EXAMCATEGORY(exam_category_code),
	CONSTRAINT REPORT_reportRateCode_fk FOREIGN KEY(report_rate_code) REFERENCES REPORTRATE(report_rate_code)
);

CREATE TABLE SUPERVISION(
	centerNo VARCHAR(10),
```

```
	examiner_code INTEGER,

	subject_paper_code VARCHAR(10),

	supervision_Date DATE,

	amountEarned DECIMAL(8,2),

	CONSTRAINT contractor_supervision_fk FOREIGN KEY(examiner_code) REFERENCES
CONTRACTOR(contractor_code),

	CONSTRAINT supervision_centerNo_fk FOREIGN KEY(centerNo) REFERENCES
EXAMCENTER(centre_NUMBER),

	CONSTRAINT supervision_subjetPaperCode_fk FOREIGN KEY(subject_paper_code) REFERENCES
SUBJECTPAPER(subject_paper_code),

);


CREATE TABLE VETTING(

	contractor_code INTEGER,

	subject_paper_code VARCHAR(10),

	number_of_script INTEGER,

	centerNo VARCHAR(10),

	dates DATE,

	CONSTRAINT VETTING_contractorCode_fk FOREIGN KEY(contractor_code) REFERENCES
CONTRACTOR(contractor_code),

	CONSTRAINT VETTING_subjetPaperCode_fk FOREIGN KEY(subject_paper_code) REFERENCES
SUBJECTPAPER(subject_paper_code),

	CONSTRAINT VETTING_centerNo_fk FOREIGN KEY(centerNo) REFERENCES
EXAMCENTER(centre_NUMBER),

);


CREATE TABLE DEPARTMENT(

	dept_id VARCHAR(5),

	dept_name VARCHAR(20),

	CONSTRAINT departmentID_pk PRIMARY KEY(dept_id),

);
```

```
CREATE TABLE WAECOFFICER(
        officer_id INTEGER,
        officerFN VARCHAR(40),
        officerMN VARCHAR(40),
        officerLN VARCHAR(40),
        phone_number VARCHAR(40),
        email VARCHAR(20),
        dept_id VARCHAR(5),
        ranks VARCHAR(20),
        username VARCHAR(20),
        password VARCHAR(20),
        CONSTRAINT subject_officer_pk primary KEY(officer_id),
        CONSTRAINT waecOfficer_department_fk FOREIGN KEY(dept_id) REFERENCES
DEPARTMENT(dept_id),
);

CREATE TABLE CLAIM(
        claim_id INTEGER IDENTITY(1,1),
        claim_type VARCHAR(20),
        contractor_code INTEGER,
        total_fee DECIMAL(8,2),
        checked_by INTEGER,
        confirmed_by INTEGER,
        approved_by INTEGER,
        clain_date DATE,
        CONSTRAINT CLAIM_claimID_pk PRIMARY KEY(claim_id),
        CONSTRAINT CLAIM_contractorCode_fk FOREIGN KEY(contractor_code) REFERENCES
CONTRACTOR(contractor_code),
```

```
	CONSTRAINT CLAIM_checkedBy_fk FOREIGN KEY(checked_by) REFERENCES
WAECOFFICER(officer_id),

	CONSTRAINT CLAIM_confirmedBy_fk FOREIGN KEY(confirmed_by) REFERENCES
WAECOFFICER(officer_id),

	CONSTRAINT CLAIM_approved_byBy_fk FOREIGN KEY(approved_by) REFERENCES
WAECOFFICER(officer_id),

);


CREATE TABLE USERS(

	users_id INTEGER,

	user_role INTEGER,

	subject_paper VARCHAR(10),

	username VARCHAR(20) UNIQUE NOT NULL,

	password VARCHAR(20) NOT NULL,

	CONSTRAINT users_userID_pk PRIMARY KEY(users_id),

	CONSTRAINT users_userID_fk FOREIGN KEY(users_id) REFERENCES
CONTRACTOR(contractor_code),

	CONSTRAINT users_subjectPaper_fk FOREIGN KEY(subject_paper) REFERENCES
SUBJECTPAPER(subject_paper_code)

);


CREATE TABLE SETUP( -- THIS TABLE SEEKS TO VALIDATE THE LENGTH OF CANDIDATE CODES

	exam_category_code VARCHAR(10),

	Cnadidate_num_length INTEGER,

	exam_paper_length INTEGER,

	CONSTRAINT setup_examCategory_fk FOREIGN KEY(exam_category_code) REFERENCES
EXAMCATEGORY(exam_category_code),

);


--MARK ENTRY

CREATE TABLE MARKENTRY(
```

```
	subject_paper_code VARCHAR(10),

	marks DECIMAL(8,2),

	keyed_count INTEGER,

	remarks VARCHAR(500),

	CONSTRAINT markEntry_subjectPaper_fk FOREIGN KEY(subject_paper_code) REFERENCES SUBJECTPAPER(subject_paper_code),

); -- THERE SHOULD BE A CHECKING THAT VALIDATES THE EXAMINER ENTERING MARKS IS THE ONE HAS BEEN ALLOCATED THE SCRIPTS


CREATE TABLE ITEMWRITING(

	subject_code VARCHAR(10),

	exam_category_code VARCHAR(10),

	subject_paper_code VARCHAR(10),

	examiner_code INTEGER,

	item_count INTEGER,

	item_rate DECIMAL(8,2),

	dates DATE,

	CONSTRAINT itemMarking_subjectCode_fk FOREIGN KEY(subject_code, exam_category_code) REFERENCES SUBJECT(subject_code,exam_category_code),

	CONSTRAINT itemMarking_examCategory_fk FOREIGN KEY(exam_category_code) REFERENCES EXAMCATEGORY(exam_category_code),

	CONSTRAINT itemMarking_subjectPaper_fk FOREIGN KEY(subject_paper_code) REFERENCES SUBJECTPAPER(subject_paper_code),

	CONSTRAINT itemMaking_examiner_fk FOREIGN KEY(examiner_code) REFERENCES CONTRACTOR(contractor_code)

);


CREATE TABLE LOGS(

	SN INTEGER IDENTITY(1,1),

	userID INTEGER,

	dates DATETIME,
```

```
        activity VARCHAR(MAX), -- TO CAPTURE ALL THE DETAILS OF ACTIVITY
        CONSTRAINT logs_serialNo_pk primary KEY(SN),
        CONSTRAINT logs_userID_fk FOREIGN KEY(userID) REFERENCES USERS(users_id),
);
```